What's New in Version 8.1

This section briefly describes what's new in version 8.1 and differences from version 8.0.

Server and Database

The Server Cluster has been implemented.
Several COM objects have been implemented to control cluster.
The server cluster can grant one or several working processes to serve client connections.
Working processes can function on one or several computers.

The 1C:Enterprise 8.1 servers cluster can operate under Linux OS management.

The 1C:Enterprise 8.1 system now works with the PostgreSQL database that operates under both Windows and Linux OS.
PostgreSQL distribution kit is included.

Compatibility with MS SQL Server 2005 has been improved.

The managed data lock mechanism has been implemented. It allows to manage data locks and enhances user parallel working in client-server mode.
The Data Lock Control Mode configuration property has been implemented.
The Data Lock Control Mode property has been implemented for the following configuration objects: catalog, document, exchange plan, chart of accounts, business process, task, chart of calculation types, chart of characteristic types, information register, accumulation register, accounting register, calculation register, recalculation, sequence and constant.
Script object DataLock has been implemented.
For BeginTransaction() global context method the parameter to set lock management mode has been implemented.

Applied Objects

Object auto numbering procedure has been changed. Numbers are obtained outside transactions without locking other transactions.

A new operating mode of the Accumulation and Accounting register managers has been implemented – with a Totals splitter – which allows a greater concurrence of writing to the registers. The GetTotalsSplittingMode() and SetTotalsSplittingMode() methods have been implemented. The Enable totals split property has been implemented for accumulation and accounting register configuration objects.

The following methods has been implemented for script objects AccumulationRegisterManager.<name> and AccountingRegisterManager.<name>: RecalcPresentTotals(), RecalcTotalsForPeriod(), SetPresentTotalsUsing(), GetPresentTotalsUsing().

The Use in Totals property has been implemented for turnover register dimensions. It allows you to exclude a dimension from the stored register totals.

The AutoPrefix property has been implemented for the Task configuration object. When a new task is created, its number is automatically supplemented with a number of the corresponding business process if the AutoPrefix property of the task is set to the BusinessProcessNumber value.

The Set Time command group has been implemented in business process list and task list extension of controls located in a table box.

The status bar displays date and time of the processed document during the interactive reposting of documents.

Query Language

The option of creating and using temporary tables in the query language has been

implemented. The TempTablesManager object and object Query property TempTablesManager have been implemented for using temporary tables. The key words have been added to query language to put data into temporary tables.

It is possible to use the configuration predefined data in the queries: enumeration values, configuration objects predefined data, empty references and business process route points values. The following system enumeration values can now be assigned to fields in the database tables: AccumulationRecordType, AccountType, AccountingRecordType.

The capabilities for specifying the periodicity using fields, for specifying several period fields, and for grouping data with a period of a second, a minute or an hour have been implemented in the Turnovers и BalanceAndTurnovers virtual tables of the accumulation and accounting registers.

Building Reports

A data composition system was developed. It allows the report generation process to be managed and reports with a complex hierarchical structure to be created. It also allows building reports based on declarative representation, manage report execution, build reports "without coding", etc.

The Report metadata object has the Data Composition Schema property.
The Report form wizard has Report Settings Form type of form and Use Data Composition System flag. The Report 1C:Enterprise script object can receive the DetailsData, SettingsCompiler and DataCompositionSchema properties. The Report 1C:Enterprise script object has the ReportResult() method.

The External Report object has been implemented. It is similar to an external data processor and it can work with the data composition system.

Script

The Global property has been implemented for the Common module configuration module.

Different non-global common modules can contain procedures/functions with same names. Non-global common module method call is done using dot after common module's name.

The Session module configuration property has been implemented. **The Session module is executed for any type of connections and is used for setting the session parameters.** The SessionParametersSetting event has been implemented for the session module.

The AddHandler and DeleteHandler script operators have been implemented. They allow to assign event handlers for applied objects, record sets and COM-objects.

Event subscription object has been implemented. Using this object you can assign event handlers for noninteractive events of one or several applied objects.

The User/Role parameter has been implemented for AccessRight() and AccessParameters() global context methods.

A ValueIsFilled() script method has been implemented.

A RandomNumberGenerator script object has been implemented.

A Title() function has been implemented in the script.

A CurrentSystemLanguage() script function has been implemented.

A new SaveUserSettings() method of the global context has been implemented. It saves user settings during the session without waiting for the application to shut down.

A new TransactionActive() global context method has been implemented. It obtains the current transaction state.

A new ErrorInfo function has been added. It allows you to obtain a full error description as structured data.
A new ShowErrorInfo method can be used to display error information programmatically.

A standby handler allows you to set delay time in fractions of a second and to set one-time activation flag in a form and global context.

The NumberInWords() and PeriodPresentation() functions allow you to set both full language names (e. g. en_US, ru_RU) and root language names (e. g. en, ru).

You can also use these functions to display sums and periods in words in Polish.

The GetCommonTemplate() method can be used both on 1C:Enterprise server and in an external connection.

The TextDocument, SpreadsheetDocument, PivotTable, ReportBuilder, PivotChart, Chart, GanttChart, Dendrogram objects became available in the 1C:Enterprise servers cluster and in the external connection module. In these modes almost all functionality not interaction actions depended is available.

Debugger

The debugging procedure has been modified significantly:

- **any 1C:Enterprise modules can now be debugged** (including sessions on the server, in external connections, and Web-services);

- multiple connections can be debugged simultaneously;
- connections can be debugged at other local machines;
- the performance measurement now displays code execution location (client or server) and server calls in specific client code strings.

Background and Scheduled Jobs

A mechanism of background and scheduled jobs has been implemented:

- background jobs allow you to initiate execution of common module procedures asynchronously (without waiting for termination);
- scheduled jobs allow you to automatically call common module procedures based on a schedule.

Integration Tools

XDTO (XML Data Transfer Objects) mechanism has been implemented - It is object data modeling mechanism. Data are described using XML schema. This mechanism is used to describe parameter types and received values of Web-services, data exchange between 1C:Enterprise 8.1 configurations with essentially different data structures and for data exchange based on XML schemas, not dependent on configurations.

The XDTO Package object and a set of script objects to work with XDTO have been implemented.

A Web-service support mechanism has been implemented.

The Web-service object has been implemented to grant web-services' functionality to configuration.

A WS-reference object has been implemented. It is intended to use external Web-services.

The web-services can be published on web-server.

The XML (DOM Level 2) document object module has been implemented.

The XML schema object module has been implemented.

The XMLWriter object has NamespaceContext property and following methods: WriteDocumentType(), WriteCDATASection(), WriteEntityReference(), WriteStartAttribute(), WriteEndAttribute(), WriteCurrent().

There are new properties and methods of reading XML, which correspond to XML Infoset.
New properties has been implemented for the XMLReader object: XMLEncoding, InputEncoding, XMLVersion, Standalone, BaseURI, HasName, SystemId, PublicId, NotationName, HasValue, IsWhitespase, IsCharasteristics, Space, Lang, NamespaceContext, IsDefaultAttribute.
New methods has been implemented for the XMLReader object: AttributeName(), AttributePrefix(), AttributeLocalName(), AttributeNamespaceURI, AttributeType(), FirstAttribute(), NextAttribute(), Skip(), MoveToContent(), FirstDeclaration(), NextDeclaration().

Script objects providing new reading/writing XML functionality have been implemented: XMLCanonicalizingWriter, XMLExpandedName, XMLExpandedNameList, XMLNamespaceContext, XMLReaderSettings, XMLWriterSettings.

XDTOSerializer which allows to serialize to/from XML all data types, stored in database and some other types has been implemented. It is intended to save object's data to XML-file and create object based on data, stored in XML-file.

Full Text Data Search

An interactive and program full-text search in data with specified search operators has been implemented (AND, OR, NO, NEAR, etc.).
The Full Text Search property has been implemented for applied objects of configuration.
Script objects have been implemented to control indexing and executing the full text search.
The FullTextSearch global context property has been implemented.
You can manage the full text data search interactively in 1C:Enterprise mode.

Common Objects

A spreadsheet document export to Excel 97 format has been implemented.

The Show() method editing has been implemented in GraphicalSchema object.

The flowchart is displayed without scroll bars when it's contents fit in the view area.
Fields grouping during editing has been implemented in the flowchart object.
The linking of connector and decorative lines to the figure center has been implemented in flowcharts.
In the flowchart lines intersections appear as arches.

The InternetMailMessage object now has the AddField() and GetField() methods.
The InternetMailMessageNonASCIISymbolsEncodingMode system enumeration has a new encoding method for non-ASCII characters — None.

You can now change message internal parameters to disable spam filter rejection of messages created by the InternetMail procedure.

The InternetMailProfile now has the following properties: POP3BeforeSMTP, POP3Authentication, SMTPAuthentication.
The InternetMailAttachment now has FileName and EncodingMode properties.
The InternetMailMessage now has the following properties: Organization, Importance, RequestDeliveryReceipt, RequestReadReceipt, Size, Categories, DeliveryReceiptAddresses, ReadReceiptAddresses.
The InternetMailText object now has the Data property.

The Convertible splitter of lines parameter and the possibility to set Encoding parameter by line containing encoding name have been implemented for TextReader, TextWriter objects constructors and for Open() methods of these objects.

The TextReader object has a new functionality of opening a file in shared? mode

The TextDocument objects' Read() and Write() methods now have: Line separator, possibility

to assign File type by line, that contains the encoding name.

You can also retrieve text from various document types and write it to a specified file in UTF-8 format. The text is retrieved by portions, i.e. you do not need to use RAM a lot.

The TextDocument object has now LineSeparator property.

The System value has been implemented in the TextEncoding system enumeration.
The 1C:Enterprise script functions StrLineCount() and StrGetLine() can recognize CR+LF, LF, CR as row splitters.

A new 1C:Enterprise script object - Range – has been implemented for setting the values range. It is used in describing the data locks.

Universal Value Collections

An Object of comparison parameter is added to the Sort() method of the ValueTable, ValueTreeRowCollection and TabularSection objects which allows sorting not by a string presentation of reference values but rather based on rules of comparing these values in the database (for example, sorting by reference for object data).
The CompareValues script object has been implemented. It allows to compare the reference values by references.

The Array and the FixedArray objects now have Find() method.

The ValueTable object has now CopyColumn() method.
The UnloadColumns() method has been implemented for tabular sections and record sets.

Indexes property support has been implemented in the ValueTable object (Indices property).

When executing value table's Find() and FindRows() methods system automatically uses index, if it meets the search conditions (if such index exists).

Interface Objects

Custom list settings now have the hierarchical view mode setting.

Document search by number in a document journal or a list is based on the list filter that is set either by the system or programmatically and cannot be changed by a user. If the search by number results in a document that does not match a user-defined filter, the system notifies a user and prompts to remove the filter.

Numerical value exchange can be copied to clipboard and then used in other programs (e. g. MS Excel, Windows Calculator) and in 1C:Enterprise formula calculator and immediate window..

If a user is asked whether it is necessary to save changes while working with Enterprise 8.1, the form with made but not saved changes is automatically located above all other forms (immediately after the question dialog box).

The Panel control now has the Picture placement property.

When the mouse cursor is moved over a text box, a tip appears for the corresponding button.

The red highlight in the required text box disappears immediately after the text box contents is changed (when a user enters data, copies data from the clipboard, selects a value from the list, etc.).

When a reference to an object marked for deletion is entered into a text box by value input by

a string, a standard warning about choosing an object marked for deletion appears.

The message text for the invalid value in the text box is modified. The message looks as follows: Incorrect data entered into the field. Press "Yes" to cancel or "No" to continue input.

When the cursor moves above the separator, the separator changes color.

The height of dropping down lists of input fields is equal 10 lines by default.

Data Exchange

Event handlers of the ExchangePlanObject.<name> OnSendNodeDataToSlave and OnReceiveNodeDataFromMaster have been implemented.

Setting Access Rights

Access restriction text templates linked to the roles have been implemented. These templates can be used for specifying the access restriction condition texts at the level of records and database fields.

The mass editing of access restriction mode at the level of records and database fields.

The rearrangement of columns, which correspond to roles, during all roles editing has been implemented.

The button to open text restrictions in new window has been implemented for all windows where access restrictions are edited. This button is available also when editing is not allowed.

Help System

A full-text search in the help system with specified search operators (AND, OR, NO, NEAR, etc.) has been implemented.

The help search dialog has been changed - the Find whole words only and Match case checkboxes have been removed. A search is performed immediately after you enter text.

Designer

A radio button with the First in group property is displayed with number 1 inside the radiobutton circle in the designer mode.

The Table box, Spreadsheet Document field, HTML Document field, Text Document field, Flowchart field, Geographical schema field controls now have the possibility to edit the Output property.

When a control is inserted in the form, it aligns even for just one existing control rather than for two or more as before.

The display of information about tables being processed during infobase converting to current version from previous version to the splash screen has been implemented.

The Common branch of the configuration tree context menu now contains the Open Common Form command.

The Move subsystem command of the Configuration window context menu has been implemented which allows the subsystem subordination to be changed within the subsystems hierarchy.

The default values of configuration objects properties has been changed: the Code (Number) type is Line value, the Code (Number) Length is equal 9, editing method - In dialog.

For the properties, which are edited in properties palette by flag, the area responding on pressing cursor is changed. Only area of flag itself is active now.

The context menu of the properties palette includes the Display properties names command.

The algorithm of a procedure generated by the print wizard has been changed.

The status indicator has been implemented in the status bar for configuration dumping to file and configuration restoring from file.

The following configuration properties has been implemented: Copyright, Provider information address, Configuration information address.

Administration

Database configuration dynamic update has been added, without an exclusive lock when changes do not affect the data structure.

The DatabaseConfigurationChangedDynamically() global context method has been implemented in order to check that the configuration has been changed dynamically during the current client session.

The event log storing structure has been changed. The log is now stored in several files.

The event log has been given the capability of setting the periodicity of the log splitting into files.

It is now possible to set the event log record in relation to the transaction (transactional or independent).

The transaction status setting has been implemented for the event log transactional records.
The filter by events, excluding events of start and finish transaction is set by default when the event log is opened.
The following fields have been added in the event log: Join, Transaction, Transaction Status, Workstation Server, General IP port, Secondary IP port.
The GetEventLogUse() and SetEventLogUse() methods have been implemented in a global context in order to check and set the registered events level.

The process of testing and repairing infobases can be divided into several sessions.
The saving of the testing and repairing parameters between the Designer work sessions has been implemented.
The IBCheckAndRepair key of launching command line now has UseStartPoint and TimeLimit parameters.

The registration of testing and repairing events in the event log has been implemented.

A mechanism for technologic logs has been implemented. It records information from all 1C:Enterprise applications, launched on one machine and is used to get dumps of abnormal application terminations and technologic problems analysis.

One can now set the infobase parameters in the Designer or using script. The following parameters are available for setting:

- Data locking time out (in seconds),
- Minimum user password length,
- Complex password.

The Cannot Change User Password property of the infobase user has been implemented.

The GetDBStorageStructureInfo() method has been implemented in global context. It allows to get 1C:Enterprise database tables and fields structure, which is used to create a storing structure in database.

One can visually analyse database locks in 1C:Enterprise server cluster management console.

It is now possible to extend the user infobase list with the common infobase lists.
One can create 1C:Enterprise 8 shortcuts to launch the infobases.
One can launch infobases and common infobases lists using shortcuts.
The RunShortcut command line parameter has been implemented.
One can save reference to infobase into file.
Editing and deleting infobases that are displayed in the list of recently used infobases have been implemented in the startup dialog.
The possibility of copying line of connection to the infobase to the clipboard has been implemented in the startup dialog.

There is a mechanism for locking the setting connections with the infobase. The setting connections lock can be set using script or using cluster management console.
The ConnectionsLock script object has been implemented.
The GetConnestionsLock() and SetConnectionsLock() methods have been implemented.
The UC command line system launch parameter and connection line parameter has been implemented.

The possibility of automatic restart of Designer after restoring infobase from file has been implemented.

The IBConnectionString executable file command line parameter has been implemented.

The users, to whom authentication by 1C:Enterprise 8.1means is not allowed, are not displayed in Select user list when connecting to the 1C:Enterprise 8.1 infobase.

The 1CEnterprise 8.1.msi installation package, which is included in distribution kit, can be used for centralized installation 1C:Enterprise 8.1 using MS Windows 2000/2003 Server group policies (in this case only English interface of setup program is possible).

You can force a break of connection with 1C:Enterprise server in cases when the server executes a code in 1C:Enterprise script or a long query to SQL Server database.

More data has been added to the cluster console. In particular, data volumes and call times are displayed for specific connections, as well as transaction locks.

The event log can be reduced while adding it to a previously saved file.
The entire reduced event log can be viewed (in case it is reduced multiple times).

The Exit() and Terminate() global context methods have the session re-start parameters.
A new StartUpSystem() method has been added to the global context.

The system attempts to re-start if the infobase is locked.

If exchange messages are loaded interactively, and a message contains configuration changes, one can launch the Designer.
Infobase re-connection attempt is now automatic.

One can use the hardware dongle version NetHASP LM 8.31.

The part of housekeeping data has been moved from C:Documents and Settings<OS User Name>Application Data1C1Cv81 to C:Documents and Settings<OS User Name>Local SettingsApplication Data1C1Cv81. Some service data have been moved from C:Documents and Settings<OS User Name>Application Data1C1Cv81 to C:Documents and Settings<OS User Name>Local SettingsApplication Data1C1Cv81.

The clone installation is not available for 1C:Enterprise 8.1. If you need to install 1C:Enterprise on several computers you can use the Active Directory directory service.

Other Modifications

The period selection form for the Day field now has the adjustment button.

The system picture library contains pictures for scheduled jobs and the data composition system.

Scrolling behavior has been changed for pages. Double clicking a page header opens / closes the page. Other pages do not change their state.

You can change background color for alternate rows in the OnRowOutput and OnDataGet event handlers.

Command line parameters to create version comparison files and configuration template files have been added.

A new ConvertFiles command line option is used to convert 1C:Enterprise 8.0 external data processors.

List table box extensions for catalogs and charts of characteristic types now have the ViewFoldersAndItems property that can display folders only, items only, or both groups and items in the table box.

The Restore Window Position command can be performed by pressing Alt + Shift + R.

The list of supported regional settings has been changed and significantly extended.

The list of supported text encodings has been significantly extended.

The Internet library has been transferred from WinInet to curl.
The proxy settings can be set by default in the inetcfg.xml file.
Setting SSL certificates for work by HTTPS protocol has been implemented in the cacert.pem file.

Optimization

The 1C:Enterprise System file-based infobases have been optimized by startup time and volume of RAM used during work with large configurations.

Metadata use is optimized for the work are loaded for the 1C:Enterprise 8 client, COM-connections, connections via Web-services.

Metadata caching at the client computer has been implemented.

Session parameters are now cached at the client computer which increases system performance when it works with configurations that have multiple calls to session parameters.

The conversion of non-periodical registers has been optimized.

A new operating mode of the accumulation and accounting register has been implemented – with a Totals splitter – which allows a greater concurrence of writing to the registers.
The possibility of deactivating of using current totals of accumulation and accounting registers.
The possibility of recalculating current totals for accumulation and accounting registers for the stated period.
The performance of writing to the accumulation and accounting registers with a large

number of changes (extra dimensions) has been increased due to changes in the database tables index structure.

The writing to accounting and accumulation registers for past periods has been optimized.

The record set writing of the accounting and accumulation registers has been optimized.

The possibility of probable deadlocks when working with the calculation registers, which support Action period, has been significantly reduced.

For the values tables line search is optimized due to possibility of indexing.

The performance of the GroupBy() method has been optimized in the values table. The search in compliance with has been optimized.

The help search has been optimized by using a full-text search (with a search time of no more than one second).

The event log reading has been optimized. The system delay has been removed.

The database configuration update process has been optimized due to a reduced time of changes analysis.

All common work of the client application is performed in a single thread that the operating system creates at the application start. It effects that speed when using COM-objects (created either by 1C:Enterprise or externally) is increased.

Document search by number in table boxes has been optimized.

Writing accumulation register record sets has been optimized for a scenario, when all or some records match the existing records of the database for the recorder.

Writing register records to accumulation and accounting registers has been optimized due to updating register totals in the privileged mode at the server.

The report builder can work better with queries that have multiple tables and fields in the selection list.

The 1C:Enterprise script works faster.

Rows in large documents are obtained faster.

Restructuring for some configuration changes (e. g. adding and deleting attributes) is performed faster.