



Document ID:	RN-CDP 2.3.1.14
Document rev:	A
Last modified:	30.07.2009
Written by:	SEL
Approved by:	RE
Product name:	CDP 2.3.1.14
Product version:	2.3.1.14
Contact:	support@icd.no

Industrial Control Design as



CDP 2.3.1.14

Release Notes

Contents

1. OVERVIEW.....	4
1.1. HIGHLIGHTS.....	4
1.2. IMPORTANT NOTICES.....	4
2. UPGRADE ISSUES.....	6
2.1. EXISTING PROJECTS.....	6
2.2. EXISTING LIBRARIES.....	6
2.3. VISUAL STUDIO 2005/2008 ISSUES	6
2.4. RTOS-32 ISSUES.....	7
2.5. MORE INFORMATION.....	7
3. CHANGE LOG.....	8
3.1. CHANGES IN LATEST VERSIONS.....	8
3.2. CHANGES IN CDP 2.3.1.7 COMPARED TO CDP 2.3.1.6.....	8
3.3. CHANGES IN CDP 2.3.1.6 COMPARED TO CDP 2.3.1.5.....	9
3.4. CHANGES IN CDP 2.3.1.5 COMPARED TO CDP 2.3.1.4.....	9
3.5. CHANGES IN CDP 2.3.1.4 COMPARED TO CDP 2.3.1.3	9
3.6. CHANGES IN CDP 2.3.1.3 COMPARED TO CDP 2.3.1.2	9
3.7. CHANGES IN CDP 2.3.1.2 COMPARED TO CDP 2.3.1.1	10
3.8. CHANGES IN CDP 2.3.1.1 COMPARED TO CDP 2.3.1.0.....	10
3.9. CHANGES IN CDP 2.3.1.0 COMPARED TO CDP 2.3.0.0.....	11
3.10. CHANGES IN CDP 2.3.0.0 COMPARED TO CDP 2.2.0.8.....	21
4. FILES IN PREVIOUS VERSIONS.....	25
5. INSTALLATION.....	26
6. KNOWN LIMITATIONS.....	26
1.1. KNOWN BUGS.....	26
7. TECHNICAL SUPPORT.....	26
8. COPYRIGHT AND LEGAL INFORMATION.....	26

1. Overview

This document describes the CDP 2.3.1 release, including the following add-ons:

- CDP On Time RTOS-32 libraries

1.1. Highlights

- Common code base for Linux, Windows and RTOS32 version
- Support for Mozilla Firefox 3.x web browsers and compatible.
- Support for the 'Qt' GUI from Trolltech
- ActivateWait functionality has changed
- RTOS version 5.14 supported from CDP version 2.3.1.8.

1.2. Important notices

Connector fix

A bug was introduced in CDP 2.3.1.8 which caused connectors to not reconnect if disconnected.

The /EHsc compile flag has changed behavior from Visual Studio 2003 to Visual Studio 2005.

See the Change log in chapter 3.

Default network interface and subnet selection in Windows

During setup the user must select the default network interface and ip-address that CDP applications will use when run under Windows. This information is stored in the registry, and CDP applications running under Windows will use this ip-address/subnet **only**.

This means that applications running under Windows will only use/see that particular subnet, even if the host computer is configured with several different ip-addresses. Previous CDP versions would receive broadcast from all the subnets configured on the computer, but would send to only one subnet, causing a faulty connection. This applies to customer created CDP applications, but also the CDPBrowsers, CDPFileManager etc.

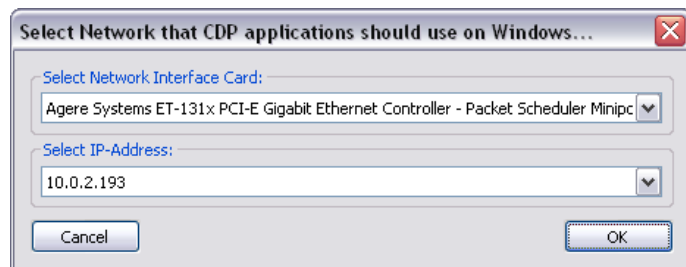


Figure 1: Default IP-address selection in CDPNetworkSetup.

To change the default ip-address/subnet used, run the *CDPNetworkSetup* tool again from the Start menu (*Start->All Programs->CDP->Tools*).

Visual Studio support

During installation the user selects whether Visual Studio 2002, Visual Studio 2003, Visual Studio 2005 (Express/Prof) or Visual Studio 2008 (Express/Prof) is to be used. Libraries for the selected version will be installed in the %CDPBase%\Libs directory. Libraries for the other Visual Studio versions will be available in the %CDPBase%\Distribution directory. A textfile 'LibraryVersion.txt' describes for which version of Visual Studio the libraries belong to. Libraries can be copied from the Distribution folder to Libs if the Visual Studio version is changed.

NOTE: All libraries that are linked together in an executable must be built using the same Visual Studio version, due to differences in the STL implementations among the Visual Studio versions.

On Time RTOS-32 support

CDP 2.3.1.14 supports On Time RTOS-32 V5.14 through the RTOS-32 add-ons.

Microsoft Window Vista support

Windows Vista is supported, but the “User Account Control” must be turned off. Unless, the WinGraphViz 3rd party tool causes a crash when used in CDPDeveloper or in the web interface.

Support for specifying Precision and Digits

Signals and Parameters can now have Precision and Digits specified in the Model file. Example XML:

```
<Properties>
  <!-- Other Properties here... -->

  <Property Name="Digits" Value="10" Description="Number of digits to show for a Signal of this type"></Property>
  <Property Name="Precision" Value="6" Description="Number of decimals to show for a Signal of this type"></Property>

  <!-- ...or here... -->
</Properties>
```

2. Upgrade issues

2.1. Existing projects

Existing projects can be used, but must be updated like this:

NOTE: All your existing project .vcproj files must be updated with `ExceptionHandling="2"` if used with Visual Studio 2005 or Visual Studio 2008. See Change log in chapter 3 for more information.

When Upgrading from CDP 2.3.0.0

- The startup code for CDP applications has been changed, so old projects need to copy/merge in the files **CDPStarter.h**, **CDPStarter.cpp** and **CDPMain.cpp** from the templates in this CDP version.
- Update the WebServer folder to get support for Opera and Mozilla Firefox web browsers. Remember to clear the cache of your web browser after upgrading to avoid problems.

Additionally, when upgrading from CDP versions prior to CDP 2.3.0.0

- **OSAPIMutex** default constructor has been hidden to force users to specify type of mutex and name during construction. OSAPIMutex instances must be initialized using constructor member initialization list in the constructor to the parent class like this:

```
MyClass::MyClass()
: myMutex(false, "MyMutexName")
{
  // constructor code
};
```

ActivateWait has changed behavior, see chapter 3.7. All projects using ActivateWait should be checked and updated.

2.2. Existing libraries

All libraries linked together in a CDP application *must* be built with the same version of CDP. Unless, the solution will refuse to build (if you're lucky), or the application will crash or behave strangely during startup or at some later time when you least expect it.

2.3. Visual Studio 2005/2008 issues

Upgrading from CDP versions prior to CDP 2.3.0.0

When compiling projects created with previous CDP versions in VS 2005/2008, the following errors will occur:

```
.\Generator\Generator.cpp(114) : error C3867: 'Generator::ProcessRunning': function call missing argument list; use '&Generator::ProcessRunning' to create a pointer to member
```

To fix, insert a `&`, then the class name and `::` in front of the function name. That is, instead of just

```
ProcessRunning
use
&Generator::ProcessRunning
```

as argument to the Register...() functions like RegisterStateProcess(), RegisterMessage() etc.

The /EHsc compile flag has changed behavior from Visual Studio 2003 to Visual Studio 2005.

- See detailed explanation in the Change log in chapter 3.

2.4. RTOS-32 issues

RTOS-32 versions

CDP 2.3.1.14 supports RTOS-32 V5.14 through the RTOS-32 add-ons.

Libraries compiled with previous versions of RTOS-32 must be rebuilt with RTOS-32 V5.14.

RTLloc configuration files (applies to project created with CDP prior to version 2.3.0.0)

The *RTLloc configuration file* CDP.cfg must be updated. Typically, two “PEHeader” lines must be added to the CDP.cfg file for your project (in the *CDP_Application* folder).

See `<CDP install dir>\Templates\Project_template\CDP_Application\CDP.cfg` for an example.

RTTBoot.com

When switching RTOS-32 version, remember that the *rttboot.com* executable on the targets must be updated to a version that matches the RTOS-32 version. The correct version of this file is found in the `<RTOS-32 installation director>\bin\rttboot.com`.

Also the *Monitor.rtb* file must be updated to to the latest version if remote debugging is to be used (this file is found in the `Debug_RTOS` folder after building a `Debug_RTOS` configuration).

RTUSB

RTOS-32 applications linked with the CDPUI add-on must also link in `rtusb.lib` from RTOS.

RTOS-32 Library order

Projects upgraded from version 4.x of RTOS-32 must be changed so that `rtt32.lib` is put last in the linker libraries, and also the `CDP*.cfg` files from templates must be copied in.

ICDBOOT.exe

CDPUI installs the file `ICDBOOT.exe` into the `%RTTarget%\bin` folder. This is done to let RTLloc have easy access to it when locating the (CDPUI) `.rtb` file. `ICDBOOT` is based on the `BIOSBOOT` boot code, and contains a modified startup-code for RTOS32 that will read the current screenmode and set it into internal structures so that CDPUI will start up with the correct screenmode. It is assumed that the program `scrmode.exe` has been run before running `rttboot CDP.rtb`. `ICDBOOT.exe` is only used in CDPUI builds.

2.5. More information

See <http://forum.icd.no> for more information and troubleshooting regarding CDP 2.3.1.14.

3. Change log

This chapter lists changes in CDP compared to previous releases.

3.1. Changes in Latest versions

NOTE: The ChangeLog has moved to Doc/ChangeLog.txt for the library in question.

The below changelog is kept for historical reasons.

3.2. Changes in CDP 2.3.1.7 compared to CDP 2.3.1.6

- Messages in component Create are written based on Application DebugLevel.
- Fix for bug 1355 (Add possibility to not print warning about too long ShortName):
Application.cpp/.h: Added new static member called m_bSupressCDPMessagesShortName, default to false. Also added new function called GetSupressCDPMessagesShortName() to get the value. The new member is set to true if "<SupressCDPMessagesShortName>1</SupressCDPMessagesShortName>" is found in Application.xml (value different than 0).
CDPObject.cpp: CDPMessage("*** Warning: %s::Create(): ShortName '%s' is too long!\n", m_fullName, m_shortName); is now only printed if the additional criterium (!Application::GetSupressCDPMessagesShortName()) is passed.
- Fix regarding bug 1345 (Handles, redundant components/objects gets different values): Configure of all redundant components is now run from VirtualApplication's Create, and resulted in a lot of CDPMessages. VirtualApplication's Create is called from the end of Application's Create, and that was before Configure had been run on CDPMessages. This resulted in using default size of LinesInHistory, and loss of the beginning of message log. A fix here, was to move the call of ConfigureCDPMessages(xmlConfig); from Application::Configure to end of Application::Create (before calling Create on Components.)
- Fix for one of the Redundancy-solutions which requires that all redundant controllers have got all information about where all signals are located inside a signal-packet. The problem was that even if all redundant controllers received the same signal-packet, and all had correct information about number of signals in the packet, not all had information about e.g. signal x should get its values in index e.g.48 in the datavalue-packet. This resulted in broken routing on web-interface for the controller missing information, and when datavalue-packets arrived, the signal-values were not updated for those signals it was missing information about. The fix is done like this: When an I/O controller builds up packets and sends them, it is now not allowed to have both redundant and non-redundant destinations. The reason why, is that e.g. the non-redundant requests signals first, and gets a packet with its values. Then the redundant requests some of the signals, and it would get the same packet, without knowing the signalnames for all positions in the packet. The new test and criterium we have added, is that the redundant recipient will discard all such packets which it doesn't know all the signalnames in.
MessengerIOserver.cpp/.h: Some new functions and changes in existing, to implement the functionality described above
- Messenger::UpdateApplication(): When received a CM_APPLICATIONNOTIFY_RD, VerifyVirtualAddress() is called already for the first companion and will set the redundantRecipient flag to true. Earlier, the flag would be set when the second companion was discovered.
- Messenger::IsRedundantHandle(int handle): New function.

3.3. Changes in CDP 2.3.1.6 compared to CDP 2.3.1.5

- Fix for bug 1350 (EventId wrong when both redundant/ non-red node on same controller): Added new static member called `ms_eventCounter_redundant`, to be used if redundant node.
 - A new function "R FIFOBuffer<T, R>::Remove(unsigned int index)" is added, which makes it possible to remove an element index-specified, not only the oldest element.
- Fixed some errors:
- R FIFOBuffer<T, R>::Remove(): Only increment `m_nTail` if `m_nLength` was ≥ 1 (if it was 0, tail would become larger than head).
 - void FIFOBuffer<T,R>::Realloc(unsigned int nNewSize): `m_nHead` was earlier changed to an incorrect value (`m_nLength` could become invalid also, if `nNewSize` was less than earlier).
 - void FIFOBuffer<T,R>::Add(R newValue): Earlier, the test was "if(`m_nHead`>`m_nBufferSize`)", and an error-message was written. This was not very likely to happen, but if it happened, `m_pBuffer[m_nHead]` was written to just after (outside array). Now the test is changed to "if(`m_nHead` \geq `m_nBufferSize`)", because index `m_nBufferSize` is illegal too. In addition, `m_nHead` is updated to a value inside array.

3.4. Changes in CDP 2.3.1.5 compared to CDP 2.3.1.4

- Fixed a bug in CDPObject that caused
- CDPSystem/EventLogger/IEventLogger.h:
Added output-parameter `timeStamp` in the abstract method `ReportSimpleEvent`.
- OSAPI/Timer/CDPTime.h:
Added new Method, `GetCurrentDateTimeMsStringCompensated(...)`.
This method returns the time in string-format, where the time is compensated by number of seconds given `i` parameter.
- CDPAAlarm/CDPAAlarm.cpp:
Retrieve `timeStamp` from `ReportSimpleEvent` and store it in `CDPAAlarm` to be used when sending `RereportSimpleEvent`, Using new `ReportSimpleEvent` with the new output parameter `timeStamp`.
- Generic/Convert.h:
Fixed `Convert` function so it rounds up to nearest if it cuts the result. (only if `Digits` ≤ 6 and `-1` \geq `number` ≤ 1)
- Fix for bug 1340 (Messenger is blocking several sec resulting in disconnect / connect):
Messenger.cpp/h:
When sending to a destination, and socket returns `ETIMEDOUT` (ARP cache is timedout), and this situation lasts for at least 20ms (new parameter in `Messenger.xml` called "MaxEtimedOutPeriod"), the Application is disconnected.
- CDPSystem/Base/CDPObject.cpp:
`CDPObject::GetComponentXML()`:
Was missing `src xml` file attribute for `TopLevelComponent` (only used in `VirtualApplication`).
This is now fixed.

3.5. Changes in CDP 2.3.1.4 compared to CDP 2.3.1.3

- When sending to a destination, and socket returns `(WSA)ETIMEDOUT` (ARP cache is timed out), and this situation lasts for at least 20ms (new parameter in `Messenger.xml` called "MaxEtimedOutPeriod"), the Application is disconnected to avoid blocking 3 ms for each message going to the 'timed out' destination.

3.6. Changes in CDP 2.3.1.3 compared to CDP 2.3.1.2

- Added support for specifying Digits and Precision for signals
- Messenger SendSocket is now made blocking with 3 ms timeout to avoid blocking messages to other applications for too long.
- RTOS32 PCI Device detection
 - Fixed a bug where if PCIList.txt contains <CR> in addition to or instead of <LF>, an exception would occur during RTOS32 startup.
 - Changed PCI Device detection to also enumerate devices on a (first-level) PCI Bridge

3.7. Changes in CDP 2.3.1.2 compared to CDP 2.3.1.1

ActivateWait behavior change

Earlier implementation of ActivateWait would activate objects in this way: If an object had ActivateWait set to 60, it was activated 60 seconds *after the first timer was initialized* (including time spent in Configure for all xml-files). This is now changed to 60 seconds after *Application::Start()* has been run.

3.8. Changes in CDP 2.3.1.1 compared to CDP 2.3.1.0

The /EHsc compile flag has changed behavior from Visual Studio 2003 to Visual Studio 2005.

CDP 2.3.1.0 (and earlier) has a bug in all Visual Studio 2005 and Visual Studio 2008 libraries and .vcproj files for Windows and RTOS. Exception Handling is set to /EHsc, which has changed behavior from Visual Studio 2003 to Visual Studio 2005. In Visual studio 2005 and later you must set the compile-flag /EHa instead of /EHsc to get the same behavior as in Visual Studio 2003/2002.

- All libraries in CDP 2.3.1.1 have been recompiled with the correct ExceptionHandling flag.
- All .vcproj files have been updated with ExceptionHandling="2", which will produce correct results in all visual studio versions.

Customers using Visual Studio 2005 and later should make sure all .vcproj files contain the attribute ExceptionHandling="2" inside the <Tool > elements:

```
<Tool
  Name="VCCLCompilerTool"
  ExceptionHandling="2"
```

```
>
```

Please note that the Linux versions of CDP do not currently support asynchronous structured exception handling. This means that division by zero, page faults and other types of illegal operations will cause the application to quit or get a Segmentation Fault.

This bug impacts the robustness/fault-tolerance of CDP applications compiled with Visual Studio 2005 and Visual Studio 2008.

3.9. Changes in CDP 2.3.1.0 compared to CDP 2.3.0.0

Changes in templates, doc and tools

ICD Bug #	Class/module	Description
	CDPFileManager	Supports WinMerge, transferfilter and support for deleting local files.
	Add-on Templates	- Now update Project Template when installed
	Add-on Demos	- Install demo projects
	Add-on Documentation	CodeDoc and User Manuals for all Add-ons. Most Add-ons also have a Programmer Manual, Release notes, and Component Test
	CDP Documentation	Updated with latest information, document template and rewritten with better explanations.

Table 1: Changes in Tools, templates, documentation etc.

Changes in the CDP library

ICD Bug #	Class/module	Description
	General	Added IEventLogger class/interface. This defines the interface for reporting events that may be picked up by an event logger, for instance the CDPEventManager.
	Application	New global variable called globalTimeAppStart storing globalTime when Application::Start() is run.
	CDPEngine	ObjectTable::FindFreeHandle(): BUGFIX, now returning 0 if no more room. Added some CDPMessages if no more room in table.
	General	Removed AlarmLogger, has been replaced by the CDPEventManager.
	General	Merged Linux branch into head. Linux now uses same codebase as the Windows and RTOS-32
	General	Added new class CDPPROPERTYDB that stores "Properties" for CDPOBJECTS and CDPModels.
	General	Added length checking and fixed NULL termination of strings wherever strcpy() was used.
	General	Added new file RLE to support run-length-encoding. Used by InputOutputStream.
	General	Added support for clock synchronization via Messenger.
	General	Added CDP libraries built for linking with DLL version of Windows run-time-libraries.
	AnalogChannel DigitalChannel	SetProperty(): Using the new and improved version of GetCDPOBJECTXML() which reads xml element name through GetProperty(), and supports shortnames which includes brackets.
	Application	When hitting 'q' on the keyboard, the keyboard handler now sends a CM_SUSPEND message to the application instead of calling Suspend directly. This is required for the 'q' key to work when the keyboard handler is called from SystemMonitorLib instead of Application::ProcessRunning.
	Application	Added WorkerTask to Application. With this one you can offload lengthy processing to the application workertask.
	Application	Added call of CDPMessages::GetInstance()->Stop() before calling OSAPIReboot() (when receiving CM_REBOOT-msg), to flush and close possible log-file.
	Application	Updated KeyboardHandling: A new member called m_bHandleKeyboardFromApplication (default true), can be set to true/false with new function EnableKeyboardHandling(), decides if HandleKeyboard() shall be called periodically. HandleKeyboard() processes registered keyboardhandlers. The CDPCoreKeyboardHandler() handles keypresses for CDP. Some "odd/special" keypresses were removed. If EnableKeyboardHandling(false) is called, the Application will not handle keypresses by calling HandleKeyboard(). Then the keyboard must be handled by another user-defined periodic process.
	Application	Moved call of OSAPISystemMonitorsInit() to CDPStarter.cpp, to make it possible to use another way to calculate CPU-load than the one in OSAPISystemMonitorsInit() (CPUMonitorStart(CPU_IDLE_TASK) and CPURelativeLoad(1000)).
	Application	:QueryTopLevelComponents() will now only return non-redundant components.
	Application	Added support for clock synchronization. 2 new parameters used to enable/disable ClockSync added to Application.xml
	Application	Added static GetInstance() that returns pointer to the one and only Application component.
984	CDPAlarm	Events (state changes and some property changes) are now logged if a component exists that implements the IEventLogger interface. Removed support for now obsolete CDPAlarmLogger.
	CDPAlarm	Added timestamp to MessageSetAlarmsListItemInfo
1187	CDPAlarm	Fixed problem with Timeout not being respected in certain cases when state changed between warning and error.
	CDPAlarm	Added support for AlarmIdentifier. FillAlarmInfo() will put this identifier in the last four bytes of the AlarmInfo message. Experimental, may be changed in the future.
	CDPAlarm	Added serialization of set, acknowledged, timestamp, m_dateTimeLastModified and EventReinsertionTimer.
	CDPAlarm	Setting LoggingEnabled="0" will now disable registering events with the local event node.

ICD Bug #	Class/module	Description
1152	CDPAlarm	Implemented new virtual override ConfigureModel() to set default value of "XMLElementName" property to "Alarm" SetProperty(): Removed handling of "Description" attribute since it is now correctly handled by the base class.
	CDPAlarm	Implemented handling of property 'Level' in GetProperty. When Level is changed, this is reported to IEventLogger in a ReportExtendedEvent with EVENT_PROPERTY_CHANGE.
	CDPAlarm	Added support for SignalInhibit in CreateXML() and SetProperty() Inhibit functionality: If inhibit signal is specified and set, the alarm will not be set. If it is already set, it will be cleared (independent of if timeout is specified)
	CDPAlarm	Changed .SignalOutset to _OutSet, .SignalOutUnack to _OutUnack, .SignalInhibit to _Inhibit and .SignalIn to _In due to name size restrictions (alarmnames would need to be very short if any of these were set).
	CDPAlarm	Fixed Serialize code (removed unnecessary serialization).
	CDPBuilder	CreateCDPObject(): Calling SetAsVirtualOwned() if parent is VirtualOwned().
	CDPBuilder	Removed SMSCommunicator and StandbyManager.
	CDPComponent	Added functionality for receiving a message requesting the component's latest set & unack'ed alarm with a specific level. Currently used by the CDPUI UISingleAlarm component.
1152	CDPComponent	Split implementation of CreateModel() in two functions, the old CreateModel(), and a new ConfigureModel(). Enables setting default values for properties in the model code and in the model.xml file. Used to enable the base classes to handle SetProperty for inherited classes even if the XML element name (tag name) is different. The XMLElementName is now read from the new property XMLElementName. Removed handling of the Activate attribute since it is now correctly handled in the base class.
	CDPComponent	CDPObject::ConfigureModel() was not called from CDPComponent::ConfigureModel().
	CDPComponent	Made SetFs() and SetModelFs() virtual.
	CDPComponent	Changed CreateComponentAndSubcomponents() to include a bool virtualOwned to get object creation to be correct for redundancy.
	CDPComponent	Added proper virtualOwned handling, and try/catch around CreateXML()
	CDPComponent	Added 'Component Suspended' Alarm which is set when Suspend() is called, and cleared when Activate() is called. Override it in XML to disable it.
348 1149	CDPConnection	Release(): Fixed reference counting to decrease reference count even if preserved is set. Added AddReference() that increases the reference count of the CDPConnection.
348 1149	CDPConnector	Added copy constructor and assignment operator. Added GetConnection() function that will increase the reference count of a connection before returning it.
	CDPConsole	Updated init function so that VMIN = 0 Removed tsetattr etc. from ::Pressed() as there is no reason for resetting and setting VMIN before and after read().
1305	CDPConsole	Fixed CTRL-C bug (Linux version).
	CDPConsole	Added support for handling arrow-keys in CDPConsole
	CDPEngine	CDPEngine is no longer global (the global engine object is removed), but created dynamically with model as specified in Application.xml. Allows overriding CDPEngine. All use of "engine." must be replaced with "CDPEngine::GetInstance()->".
	CDPEngine	Warn if we create multiple instances of this singleton class Also make sure not to leave dangling pointer when it gets destroyed.

ICD Bug #	Class/module	Description
	CDPEngine	Added possibility to print CDPMessage for every message delivered to a component by setting its debug level to 9 (DEBUGLEVEL_ALL).
	CDPEngine	Set default values for fs for the 3 CDPProcess threads. Prevents ts and thus globalTime from becoming INF.
	CDPEngine	Added more info in debug messages, like message origin on 'dumping message' printout.
	CDPEngine	Changed "Component Suspended" alarm to read "A Component is Suspended" to avoid conflict with CDPComponent.Component Suspended alarm.
	CDPEngine	Added debug message if a message is dropped due to object suspended or not system message.
	CDPEngine	Added LastObject() and GetObject() methods.
	CDPMessages	By flushing explicitly on the stdout, we can get more proper output on console, in case of segfaults and pipe/grep the output more dynamically. Only adding the flush on each line for debug. For release, only do if we hit the 10 line watermark.
	CDPMessages	Added printf in CDPMessage that gets printed if CDPMessage is executed without the message thread running.
	CDPMessages	Fixed memory allocation for message log to be less hungry
	CDPMessages	Added flushing of logfile lines to disk. Using 'c' flag on LogFiles so that the contents of the file buffer are written directly to disk if either fflush or _flushall is called. Calling fflush() in Stop() ifLogFile is used.
	CDPMessages	Printing messages with printf before CDPMessages thread is running to avoid visual delay on startup.
	CDPMessages	Fixed problems with not closing down properly.
1151	CDPModel	Added new member variable m_properties of type CDPPPropertyDB. Added new function SetProperty(). Added new function GetProperty()
	CDPNetwork	New function AddRoutingEntry() moved from UDP-classes.
1149	CDPObject	The ActivateWait property earlier wrote to the wrong object. This could lead to the Activate attribute being written to the wrong place in the component file, or even outside the file.
	CDPObject	Added new function GetProperty(). Should be the preferred function used to retrieve "Properties" from CDPObjects in the future. Returns default value set by the model if no virtual override has returned another value (see below)
1152	CDPObject	Added support for XMLElementName property. This property specifies the name of the XML element that the object uses. It should be set in ConfigureModel() for overridden models that needs to change it, and it may be overridden by specifying a different value for the property in the Model xml file. Using this property enables base-class implementations of for instance SetProperty() to handle standard properties also for overridden classes. This was not possible earlier because base classes would use hard-coded names for the XML element name.

ICD Bug #	Class/module	Description
1152	CDPObject	<p>Split CreateModel() implementation into CreateModel() and ConfigureModel(). Moved most of old CreateModel() code into ConfigureModel(). This was necessary to enable setting default values for model properties before reading possible overrides from the model file. Added new function GetModel()</p> <p>Added new function ConfigureModel(). This function is called right after CreateModel() (from CDPObject::Create() when the first instance of a model is created). Override this function to set default values for model property values returned by GetProperty(). Values specified in model XML file will override the default values specified here as long as the base implementation is called after the default values have been set. Set default values for the model properties by calling GetModel()->SetProperty("PropertyName", "PropertyValue");</p> <p>CDPObject::SetProperty(): Because of the new property implementation it is now possible to handle most standard properties in CDPObject::SetProperty(). GetProperty() is used to retrieve the name of the XML element, opposed to earlier when one had to reimplement SetProperty() in the inherited classes if the XML element name was different. Description, DisplayLevel and AccessLevel properties are now handled by CDPObject alone, they have been removed from SignalBase, Alarm, CDPParma, IOChannel etc.</p>
	CDPObject (Signal AnalogChannel DigitalChannel)	<p>Added overload of GetCDPObjectXML() that is more intelligent: Will use properties XMLElementName and XMLParentElementName to find an object's XML even if the Name attribute in XML does not equal the object's ShortName. This enables support for for instance IO Servers that have an XML format like: <Node Name="Encoder"> <Channel Name="Speed"></Channel> <.> If the object's shortname is "Encoder.Speed", the new GetCDPObjectXML() will be able to locate it if the properties XMLElementName="Channel", and XMLParentElementName="Node". This fix makes SetProperty work for such signals.</p>
	CDPObject	Fixed CDPObject::GetCDPObjectXML() to enable finding names which includes brackets ("[" and "]"). GetCDPObjectXML(std::string&, CDPXMLConfig&, CDPXMLConfig&) is deprecated. Use GetObjectXML(CDPXMLConfig& CDPXMLConfig&) instead.
	CDPObject	Made many of the getter functions const.
	CDPObject	Added CDPPropertyDB
	CDPObject	Added CM_STATUSUPDATE and CM_STATEINFO as IsSystemMessage() 'members'
	CDPObject	Added function GetAlias() to return a CDPObject alias.
	CDPObject	Added proper virtualOwned support.
1152	CDPParma	<p>Implemented new virtual override ConfigureModel() to set default value of "XMLElementName" property to "CDPParma". SetProperty(): Added support for "Value" property. Can set value in decimal or hex. Removed handling of the following attributes since it is now correctly handled by the base class: "Description", "AccessLevel", "DisplayLevel"</p>
	CDPParmaTimer	Added Serialize functions
	CDPPropertyDB	Updated GetProperty() to search for property before returning it. Added function CDPPropertyDB::Exists(). Bugfix: CDPPropertyDB::CreateXML() returned invalid xml earlier.
	CDPRampTimer	Added Serialize functions
	CDPSignalLogger	Fixed bug with iterator++ causing crash (iterated after end)
	CDPSignalLogger	Fixed problem with MaxFiles not working if not specified.
	CDPSignalLogger	The CDPSignalLogger is now documented in headerfile, runs in separate thread and is now compatible with CDPBrowser (logfiles can be read from browser).

ICD Bug #	Class/module	Description
	CDPTime	operator -=: Fixed error in code if (millitm < rhs.millitm). In the 3 functions GetTimeRelative(const char* time, const char* date), GetTimeRelative(std::string &strDosTime) and GetTimeSince(const char* givenTimeStr) there were the same error when subtracting millitm from each other: must cast to double before subtraction, millitm is unsigned, result can be negative. Added a new function GetZeroTime().
	CDPTime	Support for non-american format.
	CDPTimer	Fixed a bug in timer serialization that would cause problems when serializing from a controller with one Mhz setting to a controller with another Mhz setting.
	CDPTimer	New member delay_double (to be able to return correct value when calling CDPTimer::Delay())
1116, 1160	CDPTimerCounting	Added new class CDPTimerCounting to support timing out after a specified number of times (calls to Tikk()) instead of after a timeout based on the clock/timer. This is beneficial in redundancy code where applications must work synchronously on different controllers since the timers will never be 100% equal.
	CDPUtils	Added IsContainedIn(std::string element, std::string container, std::string separatorChar)
	CDPXMLConfiguration	ReadXMLFile(): Convert all backslashes in path to forward slashes.
	CDPXMLConfiguration	Fix to prevent GetElement() to fail when signal names contain brackets (ignore brackets within apostrophes).
	CDPXMLConfiguration	Changed writing of file to use "wbc" (commit flag) to REALLY flush file on flush().
992	CDPXMLConfiguration	Removed debug message in FindNextValue() if the element was not found. Fixed test in SaveValue() to allow xml elements with more than 1024 bytes in size. The 1024 byte limit is now on the element name, not on the element data.
1105	Convert	Fix for displaying digits - now forces all specified digits to be displayed, and possibility to choose whether exponential form is to be used.
	DigitalChannel	Tested if m_pIOServer != NULL before using it to allow using DigitalChannel also outside IOserver components (that is, in normal components).
	DispatchString	ConvertStringToSignals now support strings that are shorter than the received message Added new class StringParserReceive Updated ConvertStringToSignals() with a fix that checks if we are converting a signal that's at the end of a string. Also added check on format string in ParseString() that adds %. If not found
	DispatchString	StringParser::ConvertStringToSignals(), Handle empty string by skipping to next signal.
	EdgeTriggeredSignal	Added Serialize functions.
	FIFOBuffer	Changed return type of operator[] from R& to R since R is already the reference type.
	Filter.h	Added serialization functions to all filters Renamed BergenstadIntegrator to LeakingIntegrator
	HTTPConnectionCDP (WebServer)	Added opening of file with "wbc" (commit) and added fflush() before closing file.

ICD Bug #	Class/module	Description
	HTTPConnectionCDP (WebServer)	Fixed a bug on Windows that would cause a "?Command=Dir&Name=" to list the root folder of the application (i.e. C:\).
	InputStream	Added debug tool to get number of bytes last read/written
	InputStream	Added support for RunLength compression.
	IOServer	Added outputDisabled checking/setting in ProcessNull() so IO Servers ignoring statemachine will have correct outputdisabled set.
	IpAddress	Copy constructor: Added 0-terminating string in m_strAddress (m_strAddress will never be longer than www.xxx.yyy.zzz for IPv4). Url& Url::operator=(const char* url): Added test for max-length and (and reducing length if necessary) for nHostNameLength and nPortLength.
	IpAddress	Fixed a bug in IpAddress == operator that would actually change the content of the left side, and it would also not compare correctly.
	IpAddress	operator==(const IpAddress& rhs): Added ntohs on rhs address (was missing earlier). Removed acquiring new values for interfaceName and mac (should not change values in '=='-operator). Function made 'const'. Updating member m_strAddress everywhere address gets new value, including calling ToString() inside FromStr(). operator==(const IpAddress& rhs) is made const. Added new function GetIpAddressAsString() returning member m_strAddress.
	IpAddress	New function IsThisMulticastAddress() moved from UDP-classes.. IsThisMulticastAddress() was wrong earlier. Multicast-addresses are in range 224.0.0.0 to 239.255.255.255. The old function would return true also for addresses 240.x.x.x and up.
	IpAddress	New function IsThisMulticastAddress()
	IOChannel AnalogChannel DigitalChannel	Implemented override of new function ConfigureModel() to set default value of "XMLElementName" property to "Channel". SetProperty(): Added support for "Value" property. Can set value in decimal or hex. Removed handling of the following attributes since it is now correctly handled by the base classes: "Description", "AccessLevel", "DisplayLevel"
	IOServer	Added 'Timeout' setting for error alarm in a Channel alarm
	Messenger	Using three sendbuffers instead of one to prevent low priority thread from blocking higher priority thread in Os's that don't support priority inheritance (Linux and Windows).
	Messenger	Messenger is no longer global (the global messenger object is removed), but created dynamically with model as specified in Application.xml. Allows overriding Messenger. All use of "messenger." must be replaced with "Messenger::GetInstance()->".
	Messenger	The UDPSendBufferTop signal was updated from Normal priority. Now the signal is changed to 3 signals called UDPLowSendBufferTop, UDPNormalSendBufferTop and UDPHighSendBufferTop.
	Messenger	Added new function GetApplicationIpAddress(int applicationNumber). Added new function CreateApplicationListXml(std::string xml) Added/fixed documentation for functions GetApplicationHandle() and GetApplicationName() and GetNumberOfApplications().
	Messenger	Warn if we create multiple instances of this singleton class Also make sure not to leave dangling pointer when it gets destroyed.
	Messenger	Fixed race condition in improved buffer handling code that could lead to delayed and lost packets. This bug had been merged in from the Linux branch, and was not present in CDP2.3.0.0.
1253	Messenger	Fix for bug "Messages to (local) redundant destination is sent twice": which was a problem for CDP redundancy. In Messenger::SendMessage(), msgs to virtualApplicationHandle are now not sent directly to CDPEngine's SendMessage(), because the message will be sent by SendPacketToLocalRedundantApp() later. In SendPacketToLocalRedundantApp(), there is done the following change: Instead of calling udpSend.SendPacketTo(), now CDPEngine's SendMessage() is called because: SendPacketTo() may not work when destination IpAddress is itself.

ICD Bug #	Class/module	Description
	Messenger	Print error message if trying to send a too large message.
	Messenger	Added logics to avoid restarting RDSyncUpdate unless necessary.
	Messenger	Added extra checks on swap handle Found solution to problem with requiring network cable to run multiple CDPs on Windows without network: Use Loopback interface on Windows.
	Messenger	Fix in ApplicationConnector::SendMessage(). Will now send CM_MESSAGERECEIPT also to local redundant application. The function SendPacketToLocalRedundantApp() is virtual to make it possible to override.
	Messenger	Improvements regarding sending messages to redundant recipients and handling the receipts.
	Messenger	Fixed a bug in GetApplicationName() and VerifyApplicationConnected() code that checked on component part instead of application part.
	MessengerDefines.h	Added Print() and "=="-operator for MessageTextCommand and MessageTextCommandWithParmaSend.
	MessengerIOServer	Added CDPParma maxOutFs, used to limit signal requests on systems/HW with limited resources. The ip-stack in Linux has kernel priority and might block everything if data is sent too frequently.
	MessengerIOServer	Fix to enable sending requests for redundant signals from non-redundant signals also when the non-redundant signal is located on the Standby controller.
	MessengerIOServer	Fix to get CDP redundancy to work when we dis/reconnect standby controller.
	MessengerIOServer	Added new virtual function MakePacket() to make it possible to override packet creation.
	MessengerIOServer	Added ResetIntervalSendingSignalRequests() for cuick-connect of signals. Will do what its name says it should do.
	MessengerIOServer	Added RDPacket indicator in auto-generated XML. Fixed checking of already existing signals Fixed routing between redundant and non-redundant components
	ModbusSerialClient	Removed setting roundtrip time for all packets since it causes occasional spikes. Roundtrip time is set per packet.
	ModbusTCPClient ModbusUDPClient	Robustified code.
	ModbusUDPClient	Fixed problem with TransactionID not being correct
	ModbusUDPClient	Changed socket from non-blocking to blocking to avoid running out of DCU's in certain situations
	OSAPIEvent	Added support for RTOS and Windows in TimedWait()
	OSAPIProcess OSAPISemaphore OSAPITask OSAPIThread	Added new type OSAPITASK_HANDLE_TYPE and changed definition of OSAPITASK_ID_TYPE to differentiate between task IDs and task HANDLES. Updated functions to use the new/changed types. Now using OSAPIGetCurrentThreadId() instead of OSAPIGetCurrentThread() when a system-wide unique ID is required. OSAPIGetCurrentThread()'s return value may only be valid for the calling thread.
	OSAPIProcess	Added RTR reboot in OSAPIReboot code.
	OSAPIProcess	Added null check on handle in OSAPIWaitForThreadToExit
	OSAPIProcess	Added functions for reading process and thread priorities, handles and IDs.
	OSAPIProcess	Changed definition of OSAPITASK_ID_TYPE to always be the thread/task handle (not the ID) for all Oses.

ICD Bug #	Class/module	Description
	OSAPIProcess	OSAPIReadCPULoad() can now return own calculated values. New functions OSAPISet/GetFlagCpuLoadUpdatedFromElsewhere() and OSAPISet/GetOwnCalculatedCPULoad() used when reading out CPU-load per thread in SystemMonitorLib.
	OSAPIProcess	Added *dormant* void OSAPISleep(unsigned int us).
	OSAPIProcess	Changed priority to highest to avoid preemption on controller reboot.
	OSAPISemaphore	Added code in OSAPIMutex::Lock() and TryLock() to detect multiple locks by a single thread (Only in DEBUG builds)
	OSAPISemaphore	Moved default initialCount from Init() to the constructor. This enables users to actually use the semaphore as it should be used, and not only as a binary semaphore
	OSAPITask	Updated Start function in Linux with checks to ensure that the new thread has not exited before error check. Added printf when not started as su in Linux Removed m_bslinitialized variable that was never used
	OSAPITask	Added hThread = 0 in Delete() for Linux. This enables reuse of an OSAPITask..
	OSAPITask	Corrected possible crash bug in SetName(): Added check for max length.
	OSAPITask/ OSAPIProcess	Moved task definitions from OSAPITask.h to OSAPIProcess.h since these should use the same defines for OS-independency.
	OSAPITask	Removed RTKTerminateTask(&hThread) from OSAPITask::Delete()
1085	OSAPITask	This was a problem with setting priority before Start() when a Task was restarted without being deleted. Start() now checks that hThread is 0 before starting a new one. A new public method GetHandle() added to allow retrieving handle of the thread (Can be used along with WaitForSingleObject() to verify when a thread have stopped) Documentation for the class upgraded.
	OSAPIThread	Added failsafe mechanism in case a thread is deleted before its Main() function has ended. This might ensure graceful quits in situations where the Main() function is faulty implemented, or if it is waiting for some external event etc. This was necessary when using the CDP2Qt interface to avoid problems during quit - needs to wait 100ms for the thread to quit..
	RTOS32Hardware	Added support for loopback ethernet interface. Added drivers introduced in rtos 5.11 Using loopback interface if no other supported network interfaces found. Call atexit(CloseFileSystem) always, not only if no ethernetinterfaces are found.?
	RTOS32Hardware	Fixed a bug where the PCI device detection would install several drivers onto the same IRQ/base-address. (xn_interface_open() called several times with different devicenummer, but the same IRQ and base-address).
	SerialRTOS	Fixed possible bug in PutByte() / PutBytes(). CDP 2.3.0.0 version would convert ms to ticks twice. Shouldn't have any visible effect since 1 tick equals 1 ms in CDP.
	SerialRTOS	TxWait() will now always wait 1 ms to allow the UART to generate interrupt/update status before the status is checked.
	Signal	Added new function IsUpdated(timeout) which can be used to check if a Signal has received valid/updated values.
	Signal	Properly enumerated RouteType Added serialization of m_data (signal value) Fixed ConnectTo() for redundant signals
1189	Signal SignalBase IOChannel DigitalChannel	Moved Invert functionality from IOChannelBase/IOChannel/DigitalChannel to SignalBase/Signal. All Signals now support the Invert property. Will return 0-value if Invert property is set. Implemented by inverting value in GetWithTimestamp() and SetWithTimestamp() Note that if Invert is set, both writing and reading a signal will set/return the inverted signal, so a signal which just uses it's local m_data will not be affected in any other way than that the signal will be stored inverted. The invert will only be "visible" once the signal is routed or read/written to memory that is also read/written by other sources.

ICD Bug #	Class/module	Description
	SignalBase	Implemented override of new function ConfigureModel() to set default value of "XMLElementName" property to "Signal". SetProperty(): Added support for "Value" property. Can set value in decimal or hex. Removed handling of the following attributes since it is now correctly handled by the base classes: "Description", "Unit", "Input", "AccessLevel", "DisplayLevel"
	SimpleSerial	Added printout of SerialConfig used, and robustified reading of ClockFrequencyMhz
	SimpleSerial	Fixed recalculation of baudrate based on ClockFrequencyMhz to work on all os'es
	Sinus	Added a new parameter called UseRealTime in xml. If != 0.0, ealtime is used as time-input (m_t is then updated with globalTime). Default, m_t is updated by m_Timer.
	UdpSendReceive	Added fix for SO_TOS to work properly.
	UdpReceive UdpSend UdpSendReceive	Robustified socket code, more checks on INVALID_SOCKET
	WebServer	Added support for new webserver command GetApplicationList. Will return xml document which lists all applications that is connected to Messenger (calls new function Messenger::CreateApplicationListXml())
	WebServer	Simple "Running" state added to WebServer to avoid the "Null" state, which is confusing when debugging in the CDP console.
1255	WebServer	Fixed bug "CDP crashes in WebServer when accessing directory that exists directly".
	WebServer	Removed locking of single-buffer, since we now have separate buffer for each httpconnectionCDP (speed improvement, memory requirement increase) Fixed stop code Removed a potential bug when cdpdirectory would report out of memory or failure.

Table 2: Changes in CDP 2.3.1.14 library.

3.10. Changes in CDP 2.3.0.0 compared to CDP 2.2.0.8

Changes in templates, doc and tools

ICD Bug #	Class/module	Description
	Templates	CDP_Application.vcproj/Target.vcproj: The running of ValidateLicense is now performed in the post-build step of the CDP_Application project instead of in the pre-build step of the Target project. It is then possible to hit F5 to build and run the application in the debugger. The Target project only needs to be build to create the RTOS-32 .rtb file. CDP_Application.vcproj: Fixed RTOS32 lib order, rtt32.lib is last. CDP.cfg: Updated with PEHeader section.
Bug #643	CDPMain	Fixed RTConfig flag initialization. Flags had to be set globally before main(), Previous versions ignored the settings since RTKernel had already been initialized.
	CDPMain	Added locating startup-drive Added RF_NO_EXIT_TASK to get ethernet debugging to work. Fixed a bug where building with rtos 4.16 would cause failure because RF_NAMED_WIN32CS was not defined. Problem fixed by defining RF_NAMED_WIN32CS = 0x00000000. Added listing of libraries included in build.
	CDPCodeDoc	CDP code documentation updated.
	CDP Developer	Added validation of submitted values to prevent the user from doing invalid operations.
	CDPFileManager	Added support for using network interface from registry. Fixed possibility to cancel upon change of IPAddress. Added Sleep(20) for every 4th packet and after each file on upload to prevent controller overload. Fixed error message on send() failure, which occurs when controller has no free memory. Fixed login (no more gigantic dialog box). Fixed upload and download verification.
Bug #749	CDPNetworkSetup	Added selecting of default network interface card for CDP applications on Windows.
	CDPNetworkSetup	Fixed some problems with buttons being enabled when they should not be, and texts. Added clearing temporary internet files (IE cache).
	GetLicense	Fixed mail type so it works properly now.
	ValidateLicense ValidateTrial	Fixed return value when executable is up-to-date, bumped version
Bug #775	Web interface	Added validation of values to prevent sending invalid data from web browser, for instance using apostrophes in object descriptions etc.
Bug #743 Bug #699	Web interface	Support for Internet Explorer 7. The problem had to do with using mixed versions of MSXML.
Bug #725	Web interface	Prevent error message when clicking in Message Parameter input box.
	Web interface	Made web interface show component priority instead of model priority. "Edit Description" editor on component page fixed/renewed. May edit long html descriptions now, and component page may be updated during editing by pressing CTRL-S. MessengerIOServer page shows if packets are redundant or non-redundant. Show Min/Max attribute for CDPParmas. Removed error message when opening component page in separate window. Changed title of component page from "Controller Top Level page" to name of component.

Table 3: Changes in Tools, templates, documentation etc.

Changes in the CDP library

ICD Bug #	Class/module	Description
	AnalogChannel	Removed the second template parameter to make the code compile inn Visual Studio 2005.
Bug #788	Application	Added code to allow specifying OverrideNetworkSetup="1" inside <NetworkInterface> tag to override the ip-address selection set by CDPNetworkSetup.
	Application	Moved initialization and destruction of CDPMessages to Application constructor/destructor.
	Application	Create DCU signals only if INSTALLATION==RTOS32.
	Base64.*	Obsolete files removed.
	BitArray	Obsolete files removed.

ICD Bug #	Class/module	Description
	CDP (several files)	<p>Less information displayed during startup. DebugLevel macro is used to hide messages unless Debug property is set on the components.</p> <p>Changes to comply with ISO standard and to compile with Visual Studio 2005:</p> <ul style="list-style-type: none"> -Updated functions to comply with ISO standard (<code>_stricmp</code>, <code>_getcwd</code>, <code>_chdir</code>, <code>_getc</code> etc) -Moved declaration of some variables outside for-loop. -Added <code>#define _CRT_SECURE_NO_DEPRECATED</code> in <code>stdafx.h</code> to disable Microsoft-specific "secure" functions and moved <code>#include <stdafx.h></code> to top of files to include this define ASAP. -Added class name in front of the last parameter in a register call. <p>Replaced "messenger." with "Messenger::GetInstance()->" and "engine." with "CDPEngine::GetInstance()->" to prepare for removing the global "messenger" and "engine" objects and support overriding the Messenger class to extend it with custom functionality.</p>
	CDPCodeGenerator	CDPCodeGenerator moved to a separate lib.
Bug #736	CDPConnector	Added helper function to help sending string message with parameter.
Bug #367	CDPAlarm	<p>Removed code to write new value to xml file when calling SetProperty for Set and Unacknowledged properties.</p> <p>These properties are used for debugging and should not be written to xml file, as this would cause the alarm to be default Set, and this is normally not what we want.</p> <p>Added code to free up memory on destruction of CDP Object.</p> <p>Updated for AlarmLogger:</p> <ul style="list-style-type: none"> -Added AlarmIdentifier. -Added DateTimeLastModified() to enable date, time readout. <p>Added new function CDPAlarm::SetAlarmText() for setting/changing the Alarm-text.</p>
	CDPComponent	<p>Added code to delete states and state-transitions from a model.</p> <p>Using type double as default if Type attribute not specified for Signal (instead of refusing to create the signal).</p> <p>Added new function CreateComponentAndSubcomponents() used by VirtualApplication for Redundancy.</p>
Bug #438	CDPComponent	Enable setting the priority per component in the component.xml file (not only per model).
Bug #610	CDPComponent	<p>Requested packet frequency was dependent of fs in model file, not component file.</p> <p>This could cause signals that was created from xml only to be transferred with wrong frequency.</p>
	CDPComponent	Added fix to prevent too long Description and Unit strings from causing memory corruption in QuerySignals.
Bug #781	CDPComponent	<p>Added code to handle HTTP POST "property".</p> <p>When commands are sent to the WebServer with HTTP POST method, the WebServer will call the destination object's SetProperty method with <code>propertyName="HTTP POST"</code>.</p> <p>The parameter will be on the format "PropertyName=PropertyValue".</p> <p>This enables sending messages larger than 1024 bytes from the web browser to a component.</p>
Bug #354	CDPComponent	Remove trailing spaces and linefeeds at end of OverrideState xml value after reading it from component file.
	CDPConnection	DebugDisplayConnections(): More info displayed (when typing 'C' in console).
	CDPDefines	<p>Added</p> <pre>#define CM_APPLICATIONNOTIFY_RD 0x00030009</pre> <p>Extended <code>CM_APPLICATIONNOTIFY</code>, contains both virtual handle (origin handle) and non-rd apphandle (struct <code>MessageApplicationNotifyRD</code>). Is now used for all redundant controllers.</p>
Bug #489	CDPEngine	<p>Added new function called GetInstance() returning <code>s_instance</code>.</p> <p>Initializing <code>messageBoxTop = 0</code>; in CDPEngine constructor (the missing initialization caused crash in Messenger when dynamically created via CDPBuilder.</p> <p>Used <code>DebugLevel</code> to control debug messages.</p>
	CDPMessages	Writing error message after string instead of in front to enable reading the error message when string is too long.
	CDPObject	<p>Added test of parent component's debug level before printing debug message.</p> <p>Added code to improve cleanup of objects</p>
	CDPSignalHistory	<p>Improved history redraw.</p> <p>Fix: UIGraph history buffer allocate decreases from 20M free down to 2M free to reduce likelihood of history buffer loss.</p>
	CDPString	Removed obsolete files.

ICD Bug #	Class/module	Description
	CDPTime	Removed __timeb64 dependencies for users of CDPTime. __timeb64 only used internally to get windows and RTOS-32 system time. Added test to prevent CDPTime from being reset more than once. Added test on return value from _ctime64(). If NULL, "??:?:??:???" is returned. Added new function called GetCurrentDate() returning a string.
	CDPXMLConfiguration	Removed constraint that the newValue parameter to SaveValue() must be maximum 1024. Checking if valueName is max 1024 chars instead. Removed bogus CDPMessage about "xmlValue position not found".
	DispatchString	Fixed a bug with empty strings not being handled.
	FIFOBuffer	Removed the second template parameter to make the code compile in Visual Studio 2005.
Bug #707	HardwareInit	Changed BaseAddress from 16 bit to 32 bit.
	IOServer	Hide debug message if DebugLevel not set. Removed some debug messages completely.
	IpAddress	Hide debug message if DebugLevel not set. Removed some debug messages totally. Changed printf to CDPMessage.
	LogFileParser	Restored int instead of unsigned int.
	OSAPIOneShotTimer/ OSAPIPeriodicTimer	Added code to prevent waiting for less than one tick. Earlier, negative timeout/periods would cause undefined results.
	OSAPIOneShotTimer	ResetDelay() bug fix in conversion from ms to s. Must divide, not multiply with 1000.
	Matrix	Updated and tested.
Bug #788	Messenger	ProcessReceive() now binds to selected ip-address instead of INADDR_ANY on windows. Causes Messenger to use only one subnet, the subnet that is selected with the CDPNetworkSetup tool. RTOS-32 version must still bind to INADDR_ANY to receive broadcast traffic, but RTOS-32 does not receive broadcast from other subnets anyway so that's ok.
	Messenger	Testing on valid destination connector before using it. Discarding messages above sendBufferTop with retryCount==0. Added Print() function to struct Message and struct MessageTransportPacket. Added new function called GetInstance() returning s_instance. ApplicationConnector::Init(): added call to udpSend.SetupMulticast() if RTOS-32 and the destination address is multicast. This is necessary due to a bug in RTOS-32 (should not be necessary to join group if only want to send to a MCgroup).
	Messenger Application	Added possibility to <UseNonBlockingSendSocket>1</UseNonBlockingSendSocket> in application.xml to force Messenger send socket to be non-blocking. Default is blocking (if no value is found in Application.xml).
	MessengerDefines	Added struct prefix to defines to avoid compilation-bugs on VS .Net 2002
Bug #494	MessengerIOServer	MessengerIOOutPacket::UpdateFs(double newFs) fixed. Did not work earlier.
Bug #722	MessengerIOServer	Merged in changes from redundancy branch. Error in packet creation/RDUpdate. Redundant app started to send RDUpdate to the Standby controllers on a non-redundant packet. Added isRDSignal member in MessengerIOSignalInfo. Added functions CreateOutPacket(), InitializePacket(), CreateInPacket()..
Bug #729	MessengerIOServer	Created fix by not removing signals from in packet when packets are released, but only marking packet as "releasing". Remove signals one by one later. This prevents the MessengerIOServer::Update() from locking the controller for long periods of time when a lot of in packets are released.
Bug #770	MessengerIOServer	Removing signal from out packet when signal is deleted. Prevent MessengerIOServer from crashing if deleting a signal that exists in out packet. Made all non-static member functions virtual to enable CDP users to override them.
Bug #802	MessengerIOServer	Added ntohl() of parameter when calling SendRDSyncUpdate() from MessengerIOROutPacketUpdater::AddDelayedReply. Wrong value would be used earlier.
	ModbusSerialClient	Fixed problem with Serial not coming online if bytes are lost.
	ModbusUDPClient	Removed debug messages if debug level not set.
	OSAPIMutex	Added support for mutex of type ST_RESOURCE in RTOS-32 to support priority inheritance. Mutex type must be specified in constructor (binary=false to use resource semaphore.) The semaphore must also be given a name in the constructor (used during deadlock detection). Lock(): added input parameter which can be used to tell where the semaphore is locked from (to be used during deadlock detection). Two new functions called GetIsBinarySemaphore() and SetName(). Printing a message if a possible deadlock is detected (if timed out).

ICD Bug #	Class/module	Description
	OSAPISemaphore	Moved #include <stdafx.h> to the beginning of file to avoid lot of warnings when compiling with latest version of compiler.
	OSAPITask	New function SetRunningFlag()
	PNPID.h	Added a lo of PNP ids for PCI detection
	RTOS32Hardware	Fixed PCI Device Base Address handling, which is now correctly a DWORD instead of a Word. Updated supported network interfaces and made NE2000 the default fall back device.
	SeatexMRUSerial-IOServer	Enclosed CDPMessages in if (DebugLevel....)
	SerialIOServer	Enclosed CDPMessages in if (DebugLevel....). Added SendReceive RoundtripTime calculation into signal, and removed an CDPMessage that occurred every successful convert.
	SerialRTOS	Fixed clearing of mailbox on close, added txwait checks. Fixed proper waiting for multi drop pause.
Bug #771	Signal	Setting RouteType to routeNotSpecified in RemoveMemoryRouting() to make web interface show routing as invalid. Earlier, the web interface did not show routing as invalid when a signal had been connected and then got disconnected.
	Timer	Moved code to prevent CDPTimer from being initialized more than once.
	UdpReceive UdpSendReceive	Added code to limit number of packets stored on the incoming socket, to avoid using all DCUs. A new member called m_maxInputUDPpacketsPerSocket is added, and two new functions are added to Set and Get the value. Default value 32 defined by MAX_INPUT_UDP_PACKETS_PER_SOCKET in CDPDefines.h Added some new debug-printouts if error and DebugLevel is high enough.
	UdpReceive UdpSend	Added new function GetSocket().
	UdpSend	Added 3 new functions called AddRoutingEntry(), IsThisMulticastAddress() and SetUpMulticast() (This function was added because it was not possible to send to a MCAaddress from RTOS-32 unless joining the MCgroup. Under Windows, it was OK.).
Bug #773	UdpSend UdpSendReceive	New parameter for constructors and Init-function, called blockInfinite. If socket shall be blocking, blockInfinite decides if the socket shall block infinite (default) if destination is not present, or if the socket shall only block for a specified number of ms. Default timeout is set to 3 ms, but can be changed by SetTimeoutBlockSending(). When Init()-function is called from inside the class, Init() is now called with all parameters, to retain the settings already set in member variables.
	UdpSendReceive	Socket type changed from int to SOCKET. Added 2 new functions: GetSocket() and GetRemoteIpReceivedFrom(). ReceivePacket() and PeekPacket(): Changed parameter in rcvfrom()-call, to get remote IP address to be used in the new function GetRemoteIpReceivedFrom().
Bug #135	WebServer	Fixed output messages, added debug levels.
Bug #796 Bug #810	WebServer	SetBuffer did not check if the buffer was already large enough, so every time a delete and new is done. Added test if the buffer is already large enough and just returns true if it is.
Bug #781	WebServer	Forward all HTTP POST requests to destination object's SetProperty method.
Bug #689	WebServer	If the configured port (80) is in use, the webserver tries the next 20 ports. Auto search may be disabled by setting <DisableAutoPort>1</DisableAutoPort> in WebServer.xml.
	WebServer	Fixed caching on DownloadFile command Fixed problems with crash/hang during exit. Removed duplicate code that caused WebServer to send two messages to the destination component when forwarding messages from the web interface to an object. Changed minimum number of connections to 6 to avoid slowness in web interface. Fixed WebServer enormous memory usage by using a shared protected buffer for all threads. Removed KeepAliveTimeout parameter. Added WebServer::GetInstance() Added ChangePriority() method to change priority of all WebServer threads. Using DebugLevel macro to let debug messages depend on WebServer debug level.
	WebServer	Added support for trying next ports if default web server port is occupied.

Table 4: Changes in CDP 2.3.1.14 library.

4. Files in previous versions

The following tables list the libraries in the CDP 2.3.1.2 release. Use this information to verify that the correct version of CDP is installed. Note that libraries for all visual studio versions are available in the 'Distribution' folder where you installed CDP.

Libraries for Visual Studio .NET (2002), folder Distribution\Libs_VS70\

Date / Time	Filesize	Filename
27.11.08 11:02	11 508 742	CDP_Debug.lib
27.11.08 11:03	10 396 224	CDP_Debug_Dynamic.lib
27.11.08 11:04	11 669 038	CDP_Debug_RTOS.lib
27.11.08 11:02	10 092 134	CDP_Release.lib
27.11.08 11:04	9 128 576	CDP_Release_Dynamic.lib
27.11.08 11:05	10 241 820	CDP_Release_RTOS.lib

Libraries for Visual Studio 2003 .NET, folder Distribution\Libs_VS71\

Date / Time	Filesize	Filename
27.11.08 11:09	11 872 172	CDP_Debug.lib
27.11.08 11:10	15 163 404	CDP_Debug_Dynamic.lib
27.11.08 11:12	12 034 874	CDP_Debug_RTOS.lib
27.11.08 11:09	10 307 362	CDP_Release.lib
27.11.08 11:11	13 041 182	CDP_Release_Dynamic.lib
27.11.08 11:12	10 456 410	CDP_Release_RTOS.lib

Libraries for Visual Studio 2005/Express, folder Distribution\Libs_VS80\

Date / Time	Filesize	Filename
27.11.08 11:13	89 950 860	CDP_Debug.lib
27.11.08 11:15	90 275 004	CDP_Debug_Dynamic.lib
27.11.08 11:17	90 632 380	CDP_Debug_RTOS.lib
27.11.08 11:14	13 498 502	CDP_Release.lib
27.11.08 11:16	12 151 352	CDP_Release_Dynamic.lib
27.11.08 11:18	13 691 212	CDP_Release_RTOS.lib

Libraries for Visual Studio 2008/Express, folder Distribution\Libs_VS90\

Date / Time	Filesize	Filename
27.11.08 11:19	111 784 476	CDP_Debug.lib
27.11.08 11:21	110 907 014	CDP_Debug_Dynamic.lib
27.11.08 11:23	112 231 546	CDP_Debug_RTOS.lib
27.11.08 11:20	14 864 320	CDP_Release.lib
27.11.08 11:22	13 496 176	CDP_Release_Dynamic.lib
27.11.08 11:25	15 074 848	CDP_Release_RTOS.lib

5. Installation

Go to <http://www.icd.no/download/Download.html> to download.

Then follow the links to the right on the download page for help with setup and to download add-ons.

6. Known limitations

1.1. Known bugs

ICD Bug #	Description
Bug #829	Files in <i>Tools\ControllerSetup\Data\DiskData.zip</i> are not updated to RTOS-32 v5.12. An updated version will be available on http://www.icd.no/download/Known issues/CDP 2.3.1.1/ .

7. Technical Support

Your feedback is very important to us. To receive technical support for the products described in this document, please contact us at forum.icd.no or send mail to support@icd.no.

8. Copyright and legal information

For the latest copyright and license information, see the license agreement that was provided with the CDP installation.

EULA - End User License Agreement

NOTICE TO USER:

PLEASE READ THIS CONTRACT CAREFULLY. BY USING ALL OR ANY PORTION OF THE SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT, INCLUDING, IN PARTICULAR THE LIMITATIONS ON:

- USE CONTAINED IN SECTION 2;
- TRANSFERABILITY IN SECTION 4;
- WARRANTY IN SECTION 6 AND 7;
- AND LIABILITY IN SECTION 8.

YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU. IF YOU DO NOT AGREE, DO NOT USE THIS SOFTWARE. IF YOU ACQUIRED THE SOFTWARE ON TANGIBLE MEDIA (e.g. CD) WITHOUT AN OPPORTUNITY TO REVIEW THIS LICENSE AND YOU DO NOT ACCEPT THIS AGREEMENT, YOU MAY OBTAIN A REFUND OF ANY AMOUNT YOU ORIGINALLY PAID IF YOU: (A) DO NOT USE THE SOFTWARE AND (B) RETURN IT, WITH PROOF OF PAYMENT, TO THE LOCATION FROM WHICH IT WAS OBTAINED WITHIN THIRTY (30) DAYS OF THE PURCHASE DATE.

1. Definitions

"Software" means:

(a) all of the contents of the files, disk(s), CD-ROM(s) or other media with which this Agreement is provided, including but not limited to:

- (i) ICD or third party computer information or software;
- (ii) related explanatory written materials or files ("Documentation"); and
- (iii) icons, pictures and fonts; and

(b) upgrades, modified versions, updates, additions, and copies of the Software, if any, licensed to you by ICD (collectively, "Updates").

"Use" or "Using" means to access, install, download, copy or otherwise benefit from using the functionality of the Software in accordance with the Documentation.

"Permitted Number" means one (1) unless otherwise indicated under a valid license (e.g. volume license) granted by ICD.

"Computer" means an electronic device that accepts information in digital or similar form and manipulates it for a specific result based on a sequence of instructions.

"ICD" means Industrial Control Design AS, Aalesund, Norway.

2. Software License

As long as you comply with the terms of this End User License Agreement (this "Agreement"), ICD grants to you a non-exclusive license to Use the Software for the purposes described in the Documentation. Some third party materials included in the Software may be subject to other terms and conditions, which are typically found in a "Read Me" file located near such materials.

2.1. General Use

You may install and Use a copy of the Software on your compatible computer, up to the Permitted Number of computers.

2.2. Server Use and Distribution

2.2.1. Unless otherwise expressly permitted hereunder, no other server or network use of the Software is permitted, including but not limited to using the Software:

- (i) either directly or through commands, data or instructions from or to another computer or
- (ii) for internal network, internet or web hosting services.

2.3. Backup Copy

You may make one backup copy of the Software, provided your backup copy is not installed or used on any computer.

You may not transfer the rights to a backup copy unless you transfer all rights in the Software as provided under Section 4.

2.4 No Modification

The binaries shall not be reverse engineered or modified.

3. Intellectual Property Rights

The Software and any copies that you are authorized by ICD to make are the intellectual property of and are owned by ICD.

The structure, organization and code of the Software are the valuable trade secrets and confidential information of ICD.

The Software is protected by copyright, including without limitation by United States Copyright Law, international treaty provisions and applicable laws in the country in which it is being used.

You may not copy the Software, except as set forth in Section 2 ("Software License").

Any copies that you are permitted to make pursuant to this Agreement must contain the same copyright and other proprietary notices that appear on or in the Software. Unless specifically and expressly permitted by ICD, you agree not to modify, adapt or translate the Software.

You also agree not to reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software except to the extent you may be expressly permitted to decompile under applicable law, it is essential to do so in order to achieve operability of

the Software with another software program, and you have first requested ICD to provide the information necessary to achieve such operability and ICD has not made such information available.

ICD has the right to impose reasonable conditions and to request a reasonable fee before providing such information.

Any information supplied by ICD or obtained by you, as permitted hereunder, may only be used by you for the purpose described herein and may not be disclosed to any third party or used to create any software which is substantially similar to the expression of the Software.

Requests for information should be directed to ICD.

Trademarks shall be used in accordance with accepted trademark practice, including identification of trademarks owners' names.

Trademarks can only be used to identify printed output produced by the Software and such use of any trademark does not give you any rights of ownership in that trademark.

Except as expressly stated herein, this Agreement does not grant you any intellectual property rights in the Software and all rights not expressly granted herein are reserved by ICD.

4. Transfer

You may not rent, lease, sublicense or authorize all or any portion of the Software to be copied onto another user's computer except as may be expressly permitted herein.

You may, however, transfer all your rights to Use the Software to another person or legal entity provided that:

(a) you also transfer each this Agreement, the Software and all other software or hardware bundled or pre-installed with the Software, including all copies, Updates and prior versions, and all copies of font software converted into other formats, to such person or entity;

(b) you retain no copies, including backups and copies stored on a computer; and

(c) the receiving party accepts the terms and conditions of this Agreement and any other terms and conditions upon which you legally purchased a license to the Software. Notwithstanding the foregoing, you may not transfer education, pre-release, or not for resale copies of the Software.

5. Multiple Environment Software / Multiple Language Software / Dual Media Software / Multiple Copies/ Bundles / Updates

If the Software supports multiple platforms or languages, if you receive the Software on multiple media, if you otherwise receive multiple copies of the Software, or if you received the Software bundled with other software, the total number of your computers on which all versions of the Software are installed may not exceed the Permitted Number.

You may not, rent, lease sublicense, lend or transfer any versions or copies of such Software you do not Use.

If the Software is an Update to a previous version of the Software, you must possess a valid license to such previous version in order to Use the Update.

You may continue to Use the previous version of the Software on your computer after you receive the Update to assist you in the transition to the Update provided that:

- the Update and the previous version are installed on the same computer;
- the previous version or copies thereof are not transferred to another party or computer unless all copies of the Update are also transferred to such party or computer;
- and you acknowledge that any obligation ICD may have to support the previous version of the Software may be ended upon availability of the Update.

6. NO WARRANTY

The Software is being delivered to you "AS IS" and ICD makes no warranty as to its use or performance.

ICD AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE.

EXCEPT FOR ANY WARRANTY, CONDITION, REPRESENTATION OR TERM TO THE EXTENT TO WHICH THE SAME CANNOT OR MAY NOT BE EXCLUDED OR LIMITED BY LAW APPLICABLE TO YOU IN YOUR JURISDICTION, ICD AND ITS SUPPLIERS MAKE NO WARRANTIES CONDITIONS, REPRESENTATIONS, OR TERMS (EXPRESS OR IMPLIED WHETHER BY STATUTE, COMMON LAW, CUSTOM, USAGE OR OTHERWISE) AS TO ANY MATTER INCLUDING WITHOUT LIMITATION NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, INTEGRATION, SATISFACTORY QUALITY, OR FITNESS FOR ANY PARTICULAR PURPOSE.

7. Pre-release Product Additional Terms

If the product you have received with this license is pre-commercial release or beta Software ("Pre-release Software"), then the following Section applies.

To the extent that any provision in this Section is in conflict with any other term or condition in this Agreement, this Section shall supersede such other term(s) and condition(s) with respect to the Pre-release Software, but only to the extent necessary to resolve the conflict.

You acknowledge that the Software is a pre-release version, does not represent final product from ICD, and may contain bugs, errors and other problems that could cause system or other failures and data loss.

Consequently, the Pre-release Software is provided to you "AS-IS", and ICD disclaims any warranty or liability obligation to you of any kind.

WHERE LEGALLY LIABILITY CANNOT BE EXCLUDED FOR PRE-RELEASE SOFTWARE, BUT IT MAY BE LIMITED, ICD'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE SUM OF FIVE DOLLARS (U.S. \$5) IN TOTAL.

You acknowledge that ICD has not promised or guaranteed to you that Pre-release Software will be announced or made available to anyone in the future, that ICD has no express or implied obligation to you to announce or introduce the Pre-release Software and that ICD may not introduce a product similar to or compatible with the Pre-release Software. Accordingly, you acknowledge that any research or development that you perform regarding the Pre-release Software or any product associated with the Pre-release Software is done entirely at your own risk.

During the term of this Agreement, if requested by ICD, you will provide feedback to ICD regarding testing and use of the Pre-release Software, including error or bug reports. If you have been provided the Pre-release Software pursuant to a separate written agreement. You agree that you may not and certify that you will not sublicense, lease, loan, rent, or transfer the Pre-release Software.

Upon receipt of a later unreleased version of the Pre-release Software or release by ICD of a publicly released commercial version of the Software, whether as a stand-alone product or as part of a larger product, you agree to return or destroy all earlier Pre-release Software received from ICD and to abide by the terms of the End User License Agreement for any such later versions of the Pre-release Software.

Notwithstanding anything in this Section to the contrary, if you are located outside the United States of America, you agree that you will return or destroy all unreleased versions of the Pre-release Software within thirty (30) days of the completion of your testing of the Software when such date is earlier than the date for ICD's first commercial shipment of the publicly released (commercial) Software.

8. LIMITATION OF LIABILITY

IN NO EVENT WILL ICD OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY DAMAGES, CLAIMS OR COSTS WHATSOEVER OR ANY CONSEQUENTIAL, INDIRECT, INCIDENTAL DAMAGES, OR ANY LOST PROFITS OR LOST SAVINGS, EVEN IF AN ICD REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS, DAMAGES, CLAIMS OR COSTS OR FOR ANY CLAIM BY ANY THIRD PARTY. THE FOREGOING LIMITATIONS AND EXCLUSIONS APPLY TO THE EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION.

ICD'S AGGREGATE LIABILITY AND THAT OF ITS SUPPLIERS UNDER OR IN CONNECTION WITH THIS AGREEMENT SHALL BE LIMITED TO THE AMOUNT PAID FOR THE SOFTWARE, IF ANY.

Nothing contained in this Agreement limits ICD's liability to you in the event of death or personal injury resulting from ICD's negligence or for the tort of deceit (fraud). ICD is acting on behalf of its suppliers for the purpose of disclaiming, excluding and/or limiting obligations, warranties and liability as provided in this Agreement, but in no other respects and for no other purpose.

For further information, please see the jurisdiction specific information at the end of this Agreement, if any, or contact ICD.

9. Export Rules

You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations (collectively the "Export Laws").

In addition, if the Software is identified as export controlled items under the Export Laws, you represent and warrant that you are not a citizen, or otherwise located within, an embargoed nation (including without limitation Iran, Iraq, Syria, Sudan, Libya, Cuba, North Korea, and Serbia) and that you are not otherwise prohibited under the Export Laws from receiving the Software.

All rights to Use the Software are granted on condition that such rights are forfeited if you fail to comply with the terms of this Agreement.

10. General Provisions

If any part of this Agreement is found void and unenforceable, it will not affect the validity of the balance of the Agreement, which shall remain valid and enforceable according to its terms.

This Agreement shall not prejudice the statutory rights of any party dealing as a consumer.

This Agreement may only be modified by a writing signed by an authorized officer of ICD.

Updates may be licensed to you by ICD with additional or different terms.

This is the entire agreement between ICD and you relating to the Software and it supersedes any prior representations, discussions, undertakings, communications or advertising relating to the Software.

NOTICES:

The ICD CDP software products (CDP, CDP Developer, CDP Operation and CDPUI programs, libraries, objectcode, source code and documentation) are the proprietary property of Industrial Control Design AS, Aalesund, Norway (ICD).

The Evaluation Version of CDP Developer may not be used to create any commercial products. The evaluation period is time limited and the product shall not be used after expiry date.

By purchasing a CDP Development Tool license, the licensee acquires the rights to develop and distribute products linked with libraries delivered with ICD CDP Developer under the following conditions:

- * The product is shipped as a binary image.
It is not permitted to ship linkable object files or library files containing any software components of ICD CDP.
- * The product does not compete with any product produced by ICD to any degree.
In particular the product is not a software development product or library.
- * Proof of a legitimate license is a unique ICD License Certificate issued by ICD.
The development licence is valid one year from issue date.
- * Binary images resulting from usage of the CDP Development Tool are not time limited.
- * The license may at any one time be used only by one single developer.
- * The software is sold "as is" without any warranty as to their performance, merchantability or fitness for any purpose.
The entire risk as to the quality and performance of the software is assumed by the licensee.
- * In no event shall ICD or anyone else who has been involved in creation, production, or delivery of this software be liable for any direct, incidental or consequential damages, such as, but not limited to, loss of anticipated profits, benefits, use, or data resulting from the use of this software, or arising out of breach of any warranty.
- * The CDP Developer license does NOT grant the right to copy and use any part of the CDP Developer installation, including any of the software tools CDPBrowser, CDPFilemanger, Controller Setup Wizard, to non-licensed users or computers. A separate product named CDP Operation is provided for this purpose.

THIRD-PARTY PRODUCTS INCLUDED IN THIS RELEASE:

ADOBE SVG VIEWER

Adobe SVG Viewer is a product from ADOBE SYSTEMS INCORPORATED.
Refer to their license agreement in the installer, in
"(install dir)\Setup Files\Adobe SVG Viewer\Eula.html", or Adobe SVG Viewer's
download area at "<http://www.adobe.com/svg>" for conditions on use and
distribution of the SVG Viewer.

WINGRAPHVIZ

WinGraphviz is free software by Ood Tsen based on the GRAPHVIZ project.
It can render the dot-language to common image formats.

GRAPHVIZ

GraphViz is a project of AT&T Labs Research. It provides a collection of tools
for manipulating graph structures and generating graph layouts.