



Product:	CDPEventManagerLib
Product version:	V1.2
Document ID:	RN-CDPEventManagerLib
Doc revision:	A
Written/Appr.:	CIV / SEL
Date:	27.10.2008

Industrial Control Design AS



CDPEventManagerLib V1.2

Release Notes

The content of this document is confidential information not to be published without the consent of Industrial Control Design AS.

Industrial Control Design AS, www.icd.no, support@icd.no, forum.icd.no

1. Overview

This document describes the CDPEventManagerLib V1.2 release.

This version is compatible with CDP version 2.3.1.0. The source code version will be compatible with later CDP versions, while the binary version is compatible with this CDP version only.

This release has passed all tests as described in TS-CDPEventManagerLib.

2. Change log

This chapter lists changes included in the latest versions compared to the previous release.

2.1. Changes in V1.2

Only minor changes.

Bugs fixed:

ICD Bug #	Description
1269	<p>Deadlock in SetAlarmsCollector::MessageCM_EVENT_UPDATE We suspect this problem might is connected to building new library with updated CDPEventManagerLib, but not updated header-files for CDPEventManager. The header file was changed for SetAlarmsCollector, but this header-file was not updated on the build computer when the error was discovered. Later, after updating the header-file, the error has not occurred again. ICD Bug #1269 is still open, and set to target CDPEventManagerLib Ver.1.3.</p> <p>try/catch have been implemented in CM_EVENT_UPDATE for SetAlarmsCollector. SetAlarmsCollector is has updated the delivering of alarm list to UI. The list is now sorted with the newest alarm on top.</p>
1263	
1264	
1270	

2.2. Changes in V1.0

Bugs fixed:

ICD Bug #	Description
927	TS-CDPEventManager Test Specification.
955	Perform full System test of CDPEventManager
969	<p>Should time out all info about handles if eventProvider connector disconnects Should not try to request data from handles that is removed. The data source does not exist any more, and the info should be removed. Will now flush internal eventDBs by marking the eventCode of events from applications that disconnect with the new flag EVENT_SOURCE_OBJECT_UNAVAILABLE. This will happen in when a previously used application number is reused by a new application. (MessageCM_EVENT_SUBSCRIPTION_CONFIRM). Info about objects from this application number will also be cleared from the objectDb.</p>
1093	<p>Handle swapping confuses event providers Swapping of application handles/numbers must be handled by the CDPEventManager. Swap of handles causes events to be sent to wrong components. Now CDPConnectors are used when sending messages both ways (see bug #1093). This is fixed. At each subscription request the subscriber handle is updated. To ensure valid handle being used, the subscription request must be performed with no more than 10 second period. (To avoid events to be sent to wrong application)</p>
1158	UM-CDPEventManager
1194●	Initialise of SimpleEventDB size
1195●	Must check if event group subscription handles many groups
1198●	<p>Extended event test fails - test: 'changeAlarmTexts' ● Fixed.</p>
1200●	<p>OVERLOAD in subscriber-components - message overload and process overload in CheckExtendedEventInfoStatus ● Optimised search for missing extended events. Prevents blocking CDPEngine thread for too long periods.</p>
1201●	<p>Consistent handling of EventCode and EventGroup ● It is now possible to use 'All' as GroupName to accept all groups in the filter.</p>
1203●	<p>The average delay of from an event happens is set until it is available to the event list shall be less than 1 second. ● Tested and verified.</p>
1209	<p>Performance ● Duplicat of 1203...</p>
1212	<p>Optimize CM_EVENT_UPDATE ● Code for requesting extended events and updating EventSourceObjectDB is moved to 'CheckExtendedEventInfoStatus'. 'CheckExtendedEventInfoStatus' is called each period, and a buffer_processor is used on the SimpleEventDB to verify that all simple events with flag set for extended event, has got the extended event in the ExtendedEventDB. ● EventSourceObjectDB with properties is also updated in 'CheckExtendedEventInfoStatus' instead of in MessageCM_EVENT_UPDATE.</p>
1213	<p>Multiple groups in filter not handled in CDPEventLogger source code EventMessages.h: Increased EVENTNODE_MAX_GROUPS_PER_MESSAGE from 16 to 27. EventProvider.h: class EventProvider, added new member m_nextGroupInfoToSend, used to hold start-index of which GroupInfo to send next time in SendSubscriptionRequest(). EventProvider.cpp: EventProvider::SendSubscriptionRequest() changed: If more than EVENTNODE_MAX_GROUPS_PER_MESSAGE groups should be sent, there is not room for all in the same message. Now it will send up to EVENTNODE_MAX_GROUPS_PER_MESSAGE groups in the message, wait until next time it is called and continue to send more groups, and so on, until all groups are sent.</p>

ICD Bug #	Description
1218	<p>Sometimes two subscription requests are sent almost simultaiously from logger to node. There are now a timeout on the subscription request. When a subscription request is sent, the state is set to 'eSubscriptionStatusRequestSent' If a confirmation is not received, the state is set to 'eSubscriptionStatusConfirmationTimeout' If the confirmation is received, then the state is set to 'eSubscriptionStatusConfirmationReceived'.</p> <p>A new subscription request will not be sent from the provider until the last sent subscription request is either timeout or confirmation received.</p> <p>Still, the EventGroupInfo in each subscription will be stored to the EventProvider, and will be added to the filter immediately. But the transfere to the node will not happend until the next subscription request. Then again, the subscription requests are sent cyclically, so this should guarantie that the EventGroupInfo is transported all the way to the nodes.</p>
1219	<p>CDPEventLogger limits subscription of GroupInfo to CDPEventNode Fixed in EventProvider.cpp , AddGroupInfo methods. Removed check for max number of GroupInfo.</p>
1224	<p>EventCodeMask behaves wrong when not using EVENT_MASK_ALL Fix: CDPEventLogger.cpp, CDPEventLogger::MessageCM_EVENT_SUBSCRIPTION_REQUEST, added ntohs() on m->eventGroups[i].m_eventCodeMask in the received message.</p>
1232	<p>Make source code compatible with VS2005</p>
1234	<p>Code cleanup / comments</p>

2.3. Changes in V0.6

Bugs fixed:

ICD Bug #	Description
1093	Handle swapping confuses event providers Swapping of application handles/numbers must be handled by the CDPEventManager. Swap of handles causes events to be sent to wrong components. Now CDPCConnectors are used when sending messages both ways (see bug #1093).
1166	Use CDPCConnector when sending messages from provider to subscriber CDPCConnectors are used when sending messages both ways, and CDPCConnectors supports handle swapping.. To enable using CDPCConnectors, the MessageSubscriptionRequest had to be modified to include the name of the subscriber, as this is required to initialize the CDPCConnector.
969	Should time out all info about handles if eventProvider connector disconnects Should not try to request data from handles that is removed. The data source does not exist any more, and the info should be removed. Will now flush internal eventDBs by marking the eventCode of events from applications that disconnect with the new flag EVENT_SOURCE_OBJECT_UNAVAILABLE. This will happen in when a previously used application number is reused by a new application. Info about objects from this application number will also be cleared from the objectDb.
1136	Changing providers for a subscriber results in getting updates from both If a subscriber subscribes to provider A, then shuts down, changes provider to provider B, and then restarts, it will receive events from both as long as the subscriptions are still arrive. This is due to both subscriptions receiving the same subscription id. Thus, the subscriber will receive events from both provider A and B, thinking that they all come from B. Now the eventLogger will check that all incoming event messages comes from the node that it has actually subscribed to (in CheckMessageRevision()).
1091	Status not set to eSubscriptionStatusInactive when provider disappears. ● Status is now updated when the connector is no longer connected.
1057	Limit number of messages sent out each second/each period. (Updated) The previously created parameter "Max messages per period" is renamed to "Max messages per second", and is now implemented to calculate a limit per period per subscription.
1138	Init of CDPParma Confirmed that parameters initialized in Create() does not overwrite the values set in the component file.

2.4. Changes in V0.5

Bugs fixed:

ICD Bug #	Description
1027	Registering event in redundant controllers Implemented IEventLogger::GetInstance(handle) which uses handle to check if the source object is redundant. Used the new GetInstance() in CDPAlarm. Implemented new override CDPEventNodeRedundant and changed constructors to enable instantiating one redundant and one non-redundant node.
1057	Limit number of messages sent out each second/each period. Use the "Max messages per period" parameter in CDPEventNode and CDPEventLogger to control the number of messages sent out each period.
1116	Messages sent upon timeout of timers - conflict with redundancy Implemented CDPTimerCounting that will count number of times called instead of using the actual time. Used the new counter CDPAlarm and CDPEventLogger.
1144	Add original eventId in reprise events To make it easier to connect reprise events to the original event, the original eventId is sent along with the reprise event. NOTE: The message format for the MessageSimpleEventUpdate changed because of this, and will therefore not be compatible with earlier versions.
1145	EventCode på reprise events Now restarts the reprise timer whenever Set, Clear or Ack is called. Calculates the last event performed on the alarm and uses this eventCode in RereportSimpleEvent.

2.5. Changes in V0.4

ICD Bug #	Description
1006	Possibility to request all buffered data when just connected. When a new subscriber subscribes, the protocol will ensure that the subscriber gets all events currently in the memory database at the provider.
1125	Wrong eventId in extended events (ReportExtendedEvent) This is fixed, and the extended event gets the correct eventId.
1100	ReportExtendedEvent - Update property database first When transferring simple events and extended events, the extended events are transferred first. This reduces the possibility that the subscriber needs to request missing extended data when a simple event indicates new extended data.
1102	Update of property database when properties changes. The property database at the subscriber is now updated when a extended data with property data arrives.
1087	Extended event transfer mechanism The extended event transfer mechanism has been reviewed to assure that extended events are updated at the subscribers, even if messages are lost during transmission.
1131	Use PropertyDB when requesting the group for an eventsource for use in the filter This is fixed, making the filter work in distributed deployments.
Stress test	FIXED when a logger requests resending of events, and the events are not found in the provider, this will now be indicated as a buffer overflow. The subscriber will then know that the event is lost and unrecoverable, and not try to do another request. (To be done: A CDPAlarm will be triggered in this situation.) OLD DESCRIPTION: Some problems have been discovered during long-time stress test. When sending events fast on the network for a longer period, the CDPEventLogger seems to miss a message. In this case, the logger requests resending of events from the last received event. The CDPEventNode responds to the request, find the old event in its local database, and resend all event after this. This causes higher 'message load' than before, in addition to the stressed situation that is already generated. It seems like the logger then is likely to loose another message, and again requests an old event from the CDPEventNode. It then looks like if the situation between the CDPEventLogger and CDPEventNode gets worse, until the CDPEventLogger requests a event that is too old. In this case the CDPEventNode don't find the requested event in its local database, and indicates a lost event to the CDPEventLogger. This situation will not be a problem in normal situation. But we will check if this situation is a bug, or if it is a performance limitation.
'Level'	Support for the event source property 'Level' is implemented for CDPEventManager system and CDPAlarm.
subscriber_template	The subscriber_template implementation is only updated to handle the "Level" property. This is a general template for implementing a subscriber. (Not a part of the CDPEventManager-lib yet. Distributed as separate .cpp-file and -h-file)
SetAlarms Collector	This is a component in the CDPEventManagerLib. This is a event subscriber. The purpose for this subscriber is to subscribe events, and pass the events to a CDPUI component. This can be looked at as an example of implementing a subscriber at the current state of the CDPEventManager. In future it shall be simpler to implement subscribers. (This component is not implemented as a part of the CDPEventManager work for RR.)

Table 1 - Changes in V0.4

2.6. Changes in V0.3

ICD Bug #	Description
	Added operators for event messages.

Table 2 - Changes in V0.3

2.7. Changes in V0.2

ICD Bug #	Description
Major upgrades	Major upgrades in transport mechanisms, and logger mechanisms.

Table 3 - Changes in V1.2

3. Installation

See User Manual and Programmers Manual.

4. Known limitations

ICD Bug #	Description
1101	<p>Avoid hardcoding of Properties to be handled.</p> <p>At the moment, there is only triggered event when an alarm is Set, Acknowledged, Cleared. And also when level is changed with SetLevel inside CDPAlarm, and when "Text" or "Group" is changed.</p> <p>We need to verify that events are triggered when properties are changed. We also must verify that Acknowledge triggers an event when a single alarm is acknowledged.</p> <p>(If the event is not triggered on change, the new status will be updated on resending the alarm status as 'RepriseAlarm').</p> <p>Summary: The CDPAlarm needs to be updated to secure triggering of events at all expected status changes.</p>
1193	<p>Gloomy initialisation of EventDB in the type definitions of Simple and Extended EventDB.</p> <ul style="list-style-type: none"> In circular buffer it should be handled if the size is set to 0. When size is set to 0, the circular buffer adds events to an array that is not allocated in memory. This error is located in 'size_type next_tail()' in 'circular_buffer.h'. <pre>return (tail_+1 == array_size_) ? 0 : tail_+1;</pre> The check for 'tail_+1 == array_size_' will always fail since tail_ is initialised to 0. And 'next_tail' is used in 'push_back' in the circular_buffer.h.
1206	<p>Alarm for buffer overflow</p> <ul style="list-style-type: none"> The alarm is implemented, but the triggering of the alarm is commented out. Must be enabled and tested.
1180	<p>Missing property in CDPEventLogger property DB.</p> <ul style="list-style-type: none"> When a subscriber requests a property for an event from the logger, and the property is not found in the PropertyDB, then the logger only prints a CDPMessage that the event property was not found. This should be improved, so that the logger requests the property from the loggers providers. Typically this will be CDPEventNodes. When the CDPEventNode also don't find the property in the propertyDB, then the node must retrieve the event source properties from the event source object.
1220	<p>Delete GroupInfo when subscription time-out.</p> <ul style="list-style-type: none"> When a subscription times out, and is deleted, then the filter information from this subscription should be removed from CDPEventManager if no other subscriptions uses the same filter. The filter should only keep information for existing subscriptions.

Remaining tasks for ver.1.3 and later versions:

- 968 Possibility to acknowledge/retrieve last alarm set
- 977 DS-CDPEventManager
- 1002 Web interface should list subscriptions and groups for node/loggers
- 1089 Event property DB system
- 1090 User subscription interface to CDPEventManager system
- 1101 Avoid hardcoding of Properties to be handled.
- 1142 Must test with CDP Redundancy
- 1159 Implement SerializeIn() and SerializeOut() node and logger
- 1180 Missing property in CDPEventLogger property DB.
- 1193 Gloomy initialisation of EventDB in the type definitions of Simple and Extended EventDB..
- 1202 It shall be possible to log events generated by the SNMPManager.
- 1204 Disk Usage
- 1205 Web interface
- 1206 Alarm for buffer overflow.
- 1207 Current status request
- 1208 Alarm specific requirements
- 1216 Make AlarmEventsCollector and SetalarmsCollector inherit from CDPEventSubscriber
- 1220 Delete GroupInfo when subscription time-out.
- 1221 Terminating subscriptions
- 1225 Subscription request is ignored at startup.
- 1227 Error: Configure and maintain subscription in CDPEventLogger
- 1238 AddBufferProcessor should return pointer to bufferprocessor
- 1240 Too many equal CM_EVENT_REQUEST_EXTENDED_INFO msgs
- 1261 Definition of struct in wrong h-file
- 1279 Do not start from beginning of map when erasing a item.
- 1287 Missing filters for events in a subscription request.
- 1288 Periodic update of simple events.
- 1289 Implementation of message handlers.
- 1291 Implementation of message handlers.
- 1292 Move code to base-class for (at least subscribers) in CDPEventManager system
- 1293 MessageCM_EVENT_REQUEST_EXTENDED_INFO in IEventProvider
- 1294 MessageCM_EVENT_SUBSCRIPTION_CONFIRM in IEventSubscriber
- 1295 Potential problem in class EventSourceObjectDB::AddInfo
- 1297 Request extended event info sent to often in CheckExtendedEventInfoStatus()
- 1298 Undefined result when index is out of 'scope'... (EventProvider::GetGroupInfo)
- 1299 IEventSubscriber have no interface for deleting providers.
- 1300 Does the handle in EventSubscription in IEventProvider update on handleswitch ?

5. Technical Support

Your feedback is very important to us. To receive technical support for the products described in this document, please contact us at support@icd.no. At forum.icd.no you will also find a lot of useful information, like Frequently Asked Questions and topics regarding CDP, installation, C++, I/O, Linux, RTOS and Web.

6. Copyright and legal information

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

7. References

Reference	Document	Description
Ref 1	UM-CDPEventManagerLib	User manual
Ref 2	RS-CDPEventManagerLib	Requirements Specification
Ref 3	TS-CDPEventManagerLib	Test Specification
Ref 4	PM-CDPEventManagerLib	Programmers Manual

Table 4 - References