

CL SERVER/CLIENT

for providing alarm/event timestamps
from Programmable Logic Controllers

User Manual
Ver 1.x Rev 1.3
DR 520 10

KLINKMANN AUTOMATION
P.O. Box 38
FIN-00371 Helsinki Finland
tel. int. + 358 9 5404940
fax int. + 358 9 5413541
www.klinkmann.com

Table Of Contents

Overview	1
Communication Protocols	2
Accessing Remote Items via the CLSERVER	4
Installing the CLSERVER	5
Installing the Server	5
Licensing by using HASP HL key	7
Software license key	7
Main principles of operation	8
Getting alarm timestamps from S7 PLCs via DASSIDIRECT	8
OPC Client part communication with DASSIDIRECT and S7 PLC	8
SuiteLink/DDE Server part	9
Siemens S7 PLC alarm DB layout	10
Demo example program for Siemens S7-300 PLC	12
Getting alarm timestamps from Allen-Bradley ControlLogix PLCs via DASABCIP	15
OPC Client part communication with DASABCIP and Allen-Bradley PLC	15
SuiteLink/DDE Server part	16
ControlLogix PLC alarm data array layout	16
Demo example program for ControlLogix PLC	19
Getting alarm timestamps from Modicon PLCs via DASMBTCP	22
OPC Client part communication with DASMBTCP and Modicon PLC	22
SuiteLink/DDE Server part	23
Modicon PLC alarm area specification	24
Getting alarm timestamps from Omron PLCs via DASOmronEth	25
OPC Client part communication with DASOmronEth and Omron PLC	25
SuiteLink/DDE Server part	26
Demo example program for Omron CS1 PLC	27
Configuring the CLSERVER	28
Server Settings Command	28
OPC Server Settings Command	30
Saving of the CLSERVER Configuration File	31
Configuration File Location	31
Topic Definition Command	32
Item Names	34
Item Names for DASSIDIRECT and DASMBTCP	34
Item Names for DASABCIP	36
Item Names for DASOmronEth	37
Monitoring and Controlling Communication with DAServers	37
Using the CLServer with Suite Link and DDE Clients	38
Using the CLServer with InTouch	38
Defining the Access Names	39
Defining the Tag names	40
Monitoring the Status of Communication with InTouch	42
Notes on Using Microsoft Excel	43
Reading Values into Excel Spreadsheets	43
Troubleshooting	43
WIN.INI entries	43
Troubleshooting menu	44

CL SERVER/CLIENT

Overview

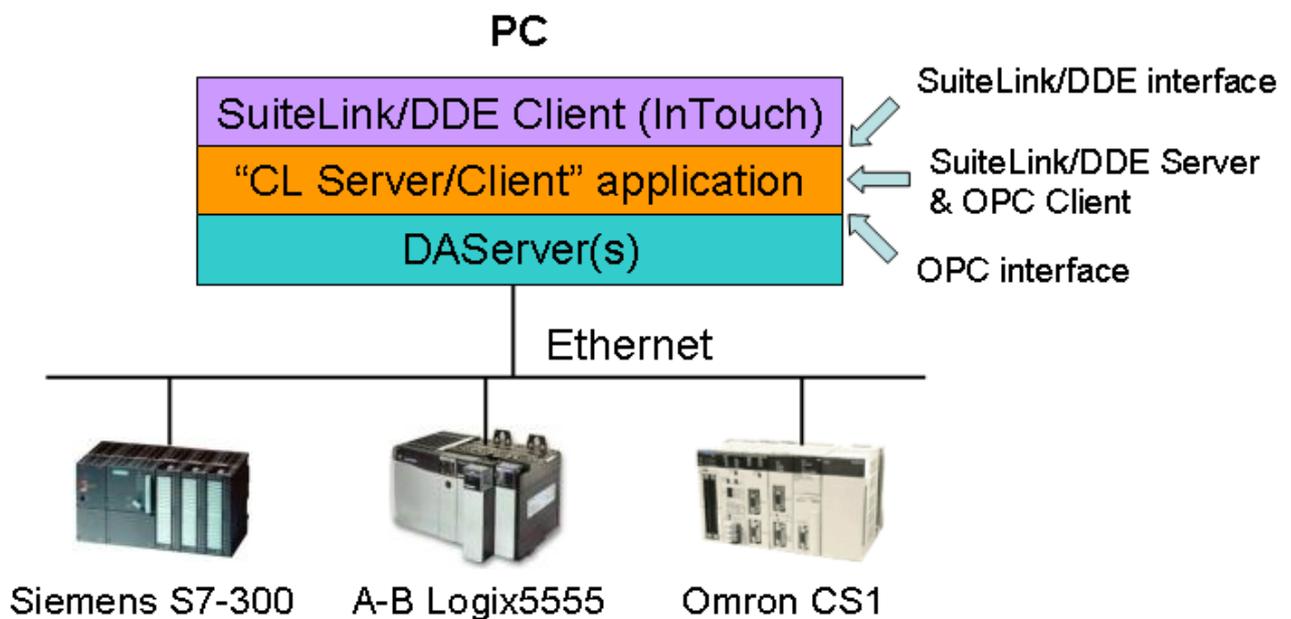
The **CL Server/Client** (“Composite Layer Server/Client“, hereafter referred to as the “CL Server/Client” or “CLSERVER” or “CLServer” or “Server”) is a Microsoft Windows 32-bit application program that acts as a communication gateway and provides other Windows application programs with alarm/event timestamping data from following Programmable Logic Controllers (PLCs):

- Siemens S7 200/300/400 family of PLCs;
- Allen-Bradley ControlLogix PLCs;
- Modicon TSX Quantum and TSX Premium;
- CS1/CJ1, CV-series and C200HX/C200HG/C200HE PLCs (directly from Ethernet network or inter-networked from other Omron networks), as well as from C-series and CQM1 PLCs (only inter-networked from other Omron networks).

The CL Server/Client correspondingly uses the following DAServers for communication with above mentiioned PLCs:

- Wonderware SIDirect DAServer (DASSIDIRECT);
- Wonderware ABCIP DAServer (DASABCIP);
- Wonderware MBTCP DAServer (DASMBTCP);
- Klinkmann Automation Omron Ethernet Direct DAServer (DASOmronEth).

In general, the system layout is like following:



To use the CL Server/Client, the appropriate additions/modifications must be done in controller program from where the alarm/event timestamping data is coming. The CL

Server/Client delivery package includes the program examples for Siemens S7-300, Allen-Bradley Logix5555 and Omron CS1 PLCs.

The CL Server/Client application consists of two main parts:

- **OPC Client** part, for connection to appropriate DAServer via OPC interface;
- **SuiteLink/DDE Server** part, providing data for SuiteLink/DDE Clients, e.g. Wonderware InTouch.

The CL Server/Client OPC Client part connects to the DAServer OPC interface when some DDE/SuiteLink client (e.g. Wonderware InTouch) connects to some of configured CL Server/Client topics, and disconnects from DAServer when all CLServer DDE/Suitelink clients are disconnected. The reading of data from DAServer is activated after activating any data item at DDE/Suitelink side and deactivated when all data items at DDE/Suitelink side are deactivated and all pending OPC transactions are completed.

If CL Server/Client OPC Client part receives new I/O item (alarm bit) activation request from SuiteLink/DDE Server part then at first this I/O item is validated - by checking item name syntax and checking presence of corresponding alarm data in the PLC.

If I/O item is valid then continuous reading of alarm status information is started to detect if there are new alarms in the PLC, i.e. CL Server/Client supports the mechanism where PLC indicates the new alarm/event occurred in PLC and CL Server/Client notifies the PLC when current alarm/event data are read from PLC and provided to SuiteLink/DDE Client.

For SuiteLink Clients (e.g. InTouch), a timestamp is provided with all alarm states delivered – as SuiteLink Clients are VTQ-aware. For DDE Clients (as they are not VTQ-aware), for each alarm it is necessary to use additional string type DDE item, where timestamp will be provided in range from “1990-1-1-0:00:00.000” to “2089-12-31-23:59:59.999”.

For "CL Server/Client" SuiteLink/DDE Server part, the alarm bit I/O item names have the exactly same syntax as DAServer bit items, for example, for DASSIDIRECT the item DB123X4.5 specifies the bit 5 in byte address 4 in DB123.

While CL Server/Client application is primarily intended for use with Wonderware FactorySuite 2000 and FactorySuite A², it may be used with any Microsoft Windows program capable of acting as a DDE, FastDDE or SuiteLink client.

Communication Protocols

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as Wonderware InTouch and Microsoft Excel can simultaneously be both a client and a server.

FastDDE provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE for the industry, this extension is being pushed to its performance constraints in distributed environments. The CL Server/Client supports the FastDDE Version 3 - an extension to Wonderware's proprietary FastDDE Version 2. This extension supports the transfer of Value Time Quality (VTQ) information. The original DDE and FastDDE Version 2 formats are still supported, providing full backward compatibility with older DDE clients. FastDDE Version 3 works on Windows 9x systems as well as Windows NT systems.

NetDDE extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM compatible computers connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

SuiteLink uses a TCP/IP based protocol and is designed by Wonderware specifically to meet industrial needs such as data integrity, high-throughput, and easier diagnostics. This protocol standard is only supported on Microsoft Windows NT 4.0 or higher. SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink was designed to be the industrial data network distribution standard and provides the following features:

- Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT operating system Performance Monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

OPC (OLE for Process Control) is an open interface standard to provide data from a data source and communicate the data to any client application in a common standard way. The OPC is based on Microsoft OLE, COM and DCOM technologies and enables simple and standardized data interchange between the industrial or office sector and the production sector. From general point of view many aspects of OPC are similar to DDE, but main difference is in the implementation by using Microsoft's COM (Component Object Model) technology. It enables fast exchange with process automation data and OPC open interface allows access to data from OPC Server in same standard way from OPC client applications supplied by different developers.

For more information on the basics of OPC, please refer to the **OPC Specification**. The OPC Data Access Custom Interface Specification is maintained by **OPC Foundation**, the current specification is 2.05a dated June 2002 (3.00 dated March 2003).

Wonderware DAServers are Wonderware's next-generation I/O servers built on Archestra™ technology. They are designed to provide simultaneous connectivity between plant floor devices and modern DDE, SuiteLink and/or OPC based client applications. DAServers support the OPC Data Access 2.05 specification and offer additional features beyond the standard, including powerful diagnostics and remote configuration capabilities.

The Suite Link, FastDDE (Version 3) and DDE support for CL Server/Client is implemented by **Wonderware I/O Server Toolkit** ver. 7,2,1,6.

The OPC support for CL Server/Client OPC Client part conforms to OPC Data Access Custom Interface Specification 2.05a.

Accessing Remote Items via the CLSERVER

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the **application name**, **topic name** and **item name**. The following briefly describes each portion of this naming convention:

application name

The name of the Windows program (server) that will be accessing the data element. In the case of alarm data coming from Siemens S7, Allen-Bradley or Omron PLCs via this Server (CL Server/Client communication gateway), the application portion of the address is **CLSERVER**.

topic name

Meaningful names are configured in the CLServer to identify specific devices (Siemens S7, Allen-Bradley or Omron PLCs). These names are then used as the topic name in all conversations to that device. For example, **PLC9**.

Note! *The topic name configured in CLServer can be same (recommended) as Device Group" name configured in Archestra System Management Console (SMC) for corresponding DAServer: in DASSIDIRECT S7CP object configuration, in DASABCIP LOGIX5000_CLX object configuration, in DASMBTCP object configuration (under TCPIP_PORT branch) or in DASOmronEth CONTROLLER object configuration; of course, any other meaningful name can be used as a topic name.*

item name

A specific data element within the specified topic. For example, when using the CLServer, items can be alarm bits in the PLC "Alarms" Data Block. The term "point" is used interchangeably with the term "item" in this User Manual. For more information on item names, see the **Item Names** section later in this manual.

Installing the CLSERVER

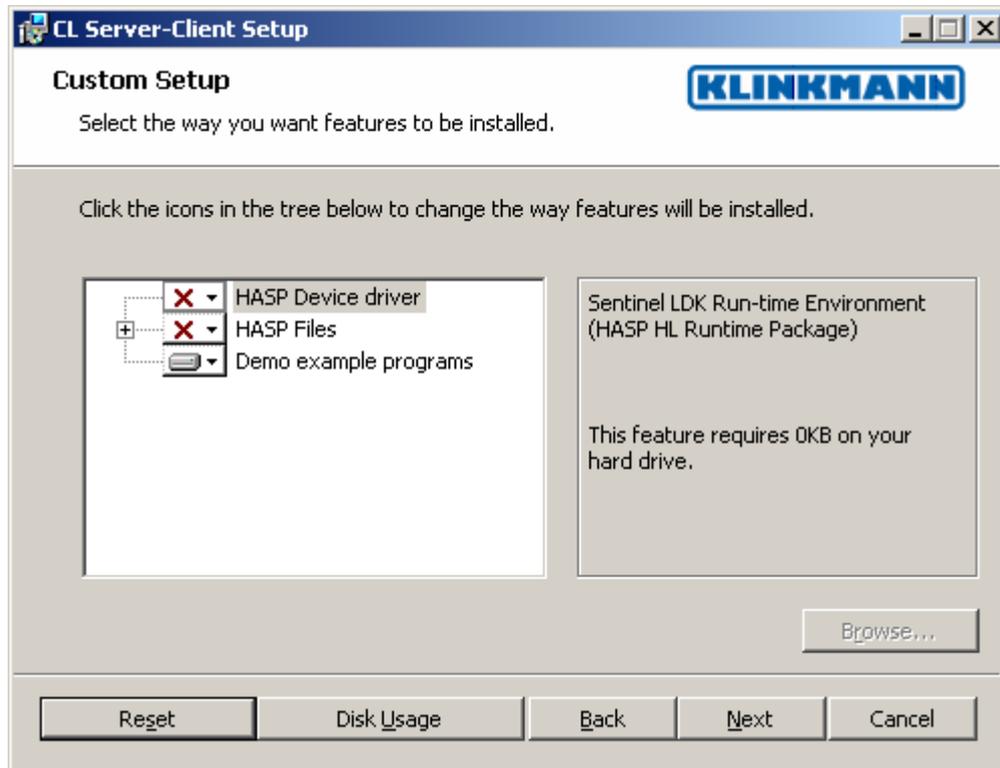
Installing the Server

The CL Server/Client installation package is supplied as a Microsoft Installer file 52010xxx.msi, where xxx is the current (latest) version of CLServer.

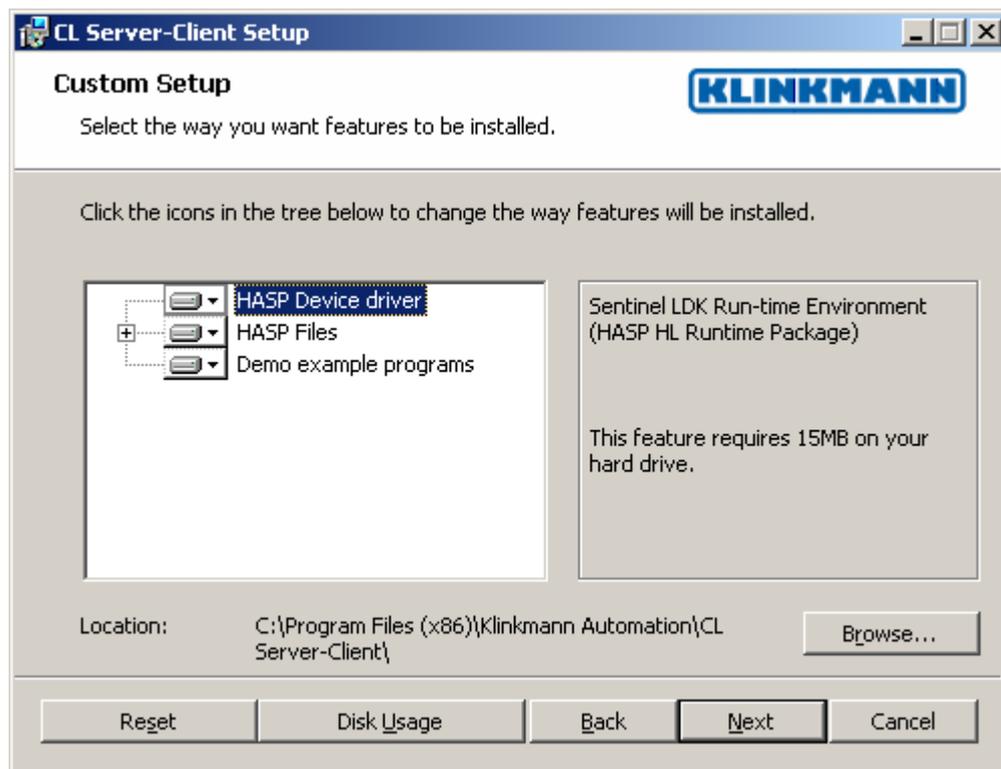
To install the CL Server/Client, run the 52010xxx.msi and proceed as directed by the CL Server/Client Setup Wizard. The installation is simple and straightforward, only it is important to select the correct protection (**HASP key** or **software license**) in “Custom Setup” dialog:

The HASP key or software license key is needed for full time running of CLServer. The **HASP key** is an USB key (dongle) to be installed into PC USB port and needs the SafeNet Sentinel LDK Run-time Environment (HASP HL Runtime Package) to be installed and running – see details in “Licensing by using HASP HL key” section below. The **software license key** is a 16-character alphanumeric “computer-dependent” string, provided after purchasing the CLServer (for more information, see “Software license key” section below. Without HASP key installed or software license key entered, the CLServer will run one hour in demo mode. After purchasing the CLServer, the appropriate HASP key or software license key is provided and no re-installation of CLServer is needed.

In case “HASP Device driver” and “HASP Files” **are not selected** then HASP USB key will not be supported and only the **software license** will be available (files needed for HASP USB key will not be installed):



In case “HASP Device driver” and “HASP Files” **are selected** then HASP USB key will be supported and both HASP-key and software license will be available (files needed for HASP USB key will be installed):



When installation is finished, the subdirectory specified as a folder where to install the CL Server/Client files will contain the following files:

CLSERVER.EXE	The CL Server/Client Program. This is a Microsoft Windows 32-bit application program.
CLSERVER.HLP	The CL Server/Client Help file.
CLSERVER.CFG	An example configuration file.
hasp_windows_4442.dll	Dynamic Link Library installed only if “HASP Files” is selected during the installation in “Custom Setup” dialog.
haspdinst.exe	Sentinel LDK Run-time Environment Installer (HASP HL Runtime Package), copied to CL Server/Client folder only if “HASP Device driver” is selected during the installation in “Custom Setup” dialog.
S7300_Demo.zip	Demo example program for Siemens S7-300 PLC
CLogix_Demo.zip	Demo example program for Allen-Bradley ControlLogix PLC
CS1_Demo.zip	Demo example program for Omron CS1 PLC
LICENSE.RTF	Klinkmann Automation software license file.

In case the “HASP Device driver” is selected during the installation in “Custom Setup” dialog, the Sentinel LDK Run-time Environment (HASP HL Runtime Package) is installed

during the CLServer installation (and will be uninstalled during the CLServer uninstallation). The presence of Sentinel LDK Run-time Environment can be checked after the CLServer installation by looking-up in Control Panel / Administrative Tools Services – the Service “Sentinel Local License Manager” must be started.

To **uninstall** the CL Server/Client, start Control Panel, select “Uninstall a program” (“Add/Remove Programs” on XP/2003) and select the “CL Server/Client” from the list of available software products. Click on “Uninstall” (“Add/Remove...” on XP/2003) and proceed as directed by the Uninstall Wizard.

Note:

*The CLServer is developed with Wonderware I/O Server Toolkit (ver 7,2,1,6) and needs the **Wonderware FS 2000 Common Components** to be installed on computer where the CLServer is running. If using Wonderware InTouch 8.0 or newer, install the FS 2000 Common Components **before** installing InTouch (see also Wonderware Tech Notes 404 and 313).*

Licensing by using HASP HL key

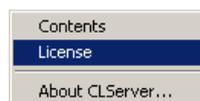
The following should be done to enable the licensing by HASP HL key:

- The “HASP Device driver” and “HASP Files” are selected during the CLServer installation in “Custom Setup” dialog – that causes correspondingly haspdinst.exe and hasp_windows_4442.dll files are copied to CLServer folder and Sentinel LDK Run-time Environment (HASP HL Runtime Package) is installed and started, enabling the CLServer can detect the HASP HL USB dongle;
- insert the received HASP key into USB port, wait until “Installing device driver software” message disappears and “Device driver software installed successfully” message appears;
- start CLServer and check - if “Software key or HaspHL key not found!” message does not appear then it means everything is done correctly and CLServer runs in full mode with licensing by HASP HL key enabled.

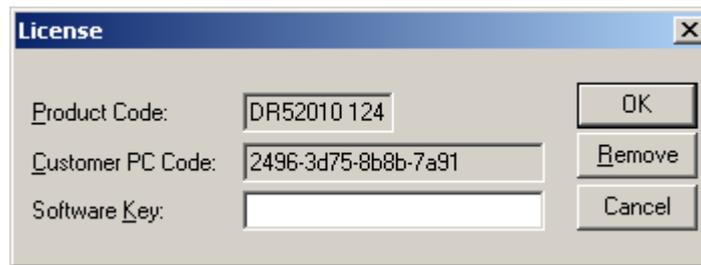
Software license key

GSM-Control supports the “computer dependent” **software licensing**. The following steps are required to enable it:

1) Start CLServer, click on "Help" menu item (also short-cut Alt+H can be used) and pop-up menu with "Help" menu commands will appear:



Select “License” and “License” dialog will appear:



2) Here the “Customer PC Code” is “computer-dependent” string generated by CLServer and it is unique for this computer. Write it down or Copy/Paste to e-mail when ordering the CLServer.

3) After purchasing the CLServer, you will get the software license key - 16-character alphanumeric string. Open the “License” dialog again and Copy/Paste it to “Software Key” field:



4) Click OK and restart CLServer. CLServer software license now is enabled.

Note – the “Software Key” string is saved to MS Windows system directory (e.g. C:\Windows) WIN.INI file [CLServer] section to enable it is automatically detected at CLServer next start-up.

Main principles of operation

Getting alarm timestamps from S7 PLCs via DASSIDIRECT

OPC Client part communication with DASSIDIRECT and S7 PLC

The CL Server/Client OPC Client part connects to the DASSIDirect OPC server when some DDE/SuiteLink client connects to the configured CLServer topic, and disconnects from DASSIDirect when all CLServer DDE/Suitelink clients are disconnected. The reading of data from DASSIDirect is activated after activation any of data items and deactivated when all data items are deactivated and all pending OPC transactions are completed.

If OPC Client part receives new I/O item (alarm bit) activation request from SuiteLink/DDE Server part then following is done:

1) Checking if received I/O item(s) is valid – by checking item name syntax and reading and checking corresponding data block configuration double word (dword), if not yet already checked by other I/O item of this datablock. The checking of configuration dword is done after the item activation, so if corresponding data block does not exist or contains invalid data, then following happens:

- notification is logged to LogViewer,
- ERRORITEMS system item value is increased,

- item is marked as bad and have bad value quality.
No further activity of this item is possible except deletion.

If configuration dword content is successfully read and has correct values then CLServer activates OPC conversations with corresponding data block. Contents of corresponding DB configuration dword indicates number of alarm dwords, alarm data block version number, alarm data block identification number and new alarms presence flag. One alarm dword can handle 32 alarm bits.

CLServer can receive item names pointing to different DBs and (or) PLCs - several alarm DBs in several PLCs (topics).

2) If I/O item is valid, the continuous reading of MSB is started. MSB indicates if changes has been occurred in alarm words bit statuses (=some bit(s) has changed 0->1 or 1->0).

3) If DB MSB (used by PLC for new alarm/event indication) is ON (non-zero) then depending on the configuration of CLServer topic:

a) if complete data block reading is configured then OPC Client part requests reading of full DB, including: alarm double words = according the configuration dword low word part and timestamps = according the amount of alarm double words;

b) if complete data block reading is not configured then OPC Client part requests reading of alarm double words = according the configuration dword low word part; in this case timestamps are read ONLY for alarms with changed status.

Both for a) and b):

- all changed alarm bit and / or timestamp values and corresponding VTQ data is transferred to CLServer SuiteLink/DDE Server part and correspondingly to SuiteLink/DDE Client (InTouch);

- OPC Client resets the configuration dword MSB and waits while OPC server will notify about MSB value change;

Corresponding DASSIDIRECT item names used internally by CLServer to obtain date and time from DASSIDIRECT, have suffix „DT” and data type = VT_BSTR.

4) If DB configuration dword MSB is OFF (zero) then it means there are no new alarms/events (=changes) in this DB, no special activity is needed

5) CL Server/Client OPC Client part detects DB number or DB's from where timestamped alarm data is available according to item name requested by SuiteLink/DDE client (for example, InTouch).

SuiteLink/DDE Server part

From SuiteLink/DDE Client (e.g. InTouch) point of view, the accessing to CL Server/Client SuiteLink/DDE Server part is completely same as accessing to DASSIDIRECT, i.e. same topic names and I/O point naming syntax is used, the only difference is the Server application name is “CLServer”.

For CL Server/Client SuiteLink/DDE Server part, the alarm bit I/O item names have the exactly same syntax as DASSIDIRECT bit items, for example, DB123X4.5 specifies the bit 5 in byte address 4 (in first alarms dword there are bytes 4...7) in DB123.

CLServer offers additional (system) items to the SuiteLink/DDE clients, see *Monitoring and Controlling Communication with DAServers* section below in this manual.

Siemens S7 PLC alarm DB layout

Configuration dword specification:

- configuration dword is 32 bit long double word (bits 0...31)
- bit 31 = MSB used for change detection
- bit 30 = reserved for future use
- bit 29 = reserved for future use
- bit 28-24 of high word high byte are reserved for version control
 - o version number is between 1-31
 - o by using the version number, CLServer can detect the alarm DB structure (this possibility is reserved for future changes of alarm DB structure)
- high word low byte has a fixed value „AA” (in hex)
 - o AA means that this DB is used for alarms
 - o This only confirms that this is really an alarm DB
- low word (bits 8..15) reserved for future use
- low word (bits 0..7) are used to inform amount of alarm double words; according to this information CLServer calculates the total space needed for alarm double words and date×tamps

Siemens S7 PLC alarm DB layout, version 1:

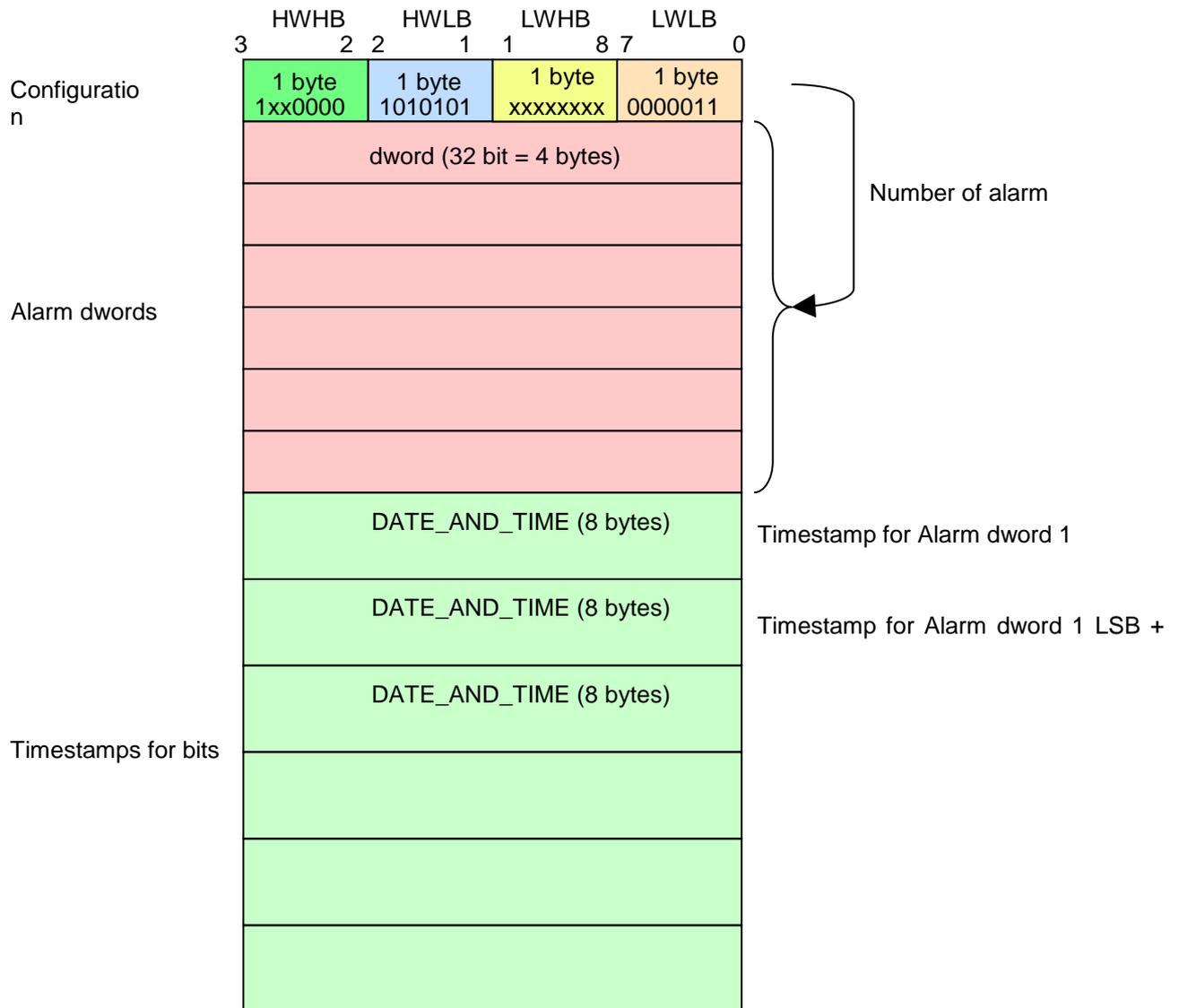
Configuration dword high word high byte (HWHB) MSB is used to indicate a change in the alarm dwords has occurred

Configuration dword high word high byte (HWHB) 5 least significant bits are used to indicate the version number of the alarm DB structure (range 1...31)

Configuration dword high word low byte (HWLB) contains the AlarmDB identification number, hex AA (10101010)

Configuration dword low word high byte (LWHB) is not used (reserved for future use)

Configuration dword low word low byte (LWLB) is used to indicate the number of alarm dwords in the DB



The size of the alarm DB (in bytes) is: $4 + LWLB * 4 + LWLB * 8 * 32$

Demo example program for Siemens S7-300 PLC

CL Server/Client distribution package contains the “Demo example program for Siemens S7-300 PLC” (**S7300_Demo.zip**). This demo program can be used as an example how to add appropriate additions/modifications in S7 PLC program from where the alarm/event timestamping data is coming.

The S7-300 PLC has to support the alarm data block structure, described in the chapter “Configuration dword specification” (see above). Therefore, a STEP 7 demo project for S7-300 PLC is provided, containing the function block ALARMMNG (FB10). The FB ALARMMNG handles the alarm data block and alarm queues.

The demo project has a following call trees:

```
CYCLE_MAIN (OB1)
  DEMO (FB1)
    ALARMMNG (FB10)
```

```
CYC_INT5 (OB35)
  ALARMGEN (FB2)
```

```
CYCL_FLT (OB80)
```

The alarms, for demonstration purposes, are generated by the FB ALARMGEN (FB2) called from the OB CYC_INT5 (OB35) every 10 ms. FB ALARMGEN changes alarm bits by shifting left the contents of alarm double words (32 alarm bits) at the specified time intervals. The FB ALARMGEN provides 10 alarm double words. The default time intervals for them are: 300, 120, 60, 30, 20, 10, 5, 3, 2, 1 seconds.

The FB ALARMMNG (FB10) is called on every PLC scan cycle. The FB call has a following format:

```
ALARMMNG.DB10 (
  AlmDBNum:=AlarmDBNum,    // Alarm data block number
  AlmDWNNum:=AlarmDWNNum,  // Alarm double word number
  AlmDW    :=AlarmDW,      // Alarm DW (32 alarm bits)
  UseExtTS:=AlarmUseExtTS, // "Use external timestamp" flag
  ExtTS    :=AlarmExtTS,   // External timestamp
  Init     :=AlarmInit);  // "Initialize Alarm Manager" flag
```

Input parameters

AlmDBNum Alarm data block (DB) number. The alarm DB contains the configuration double word (DW), alarm DWs and alarm bit timestamps, as described in the chapter “Configuration word specification”.

AlmDWNNum Alarm DW number within the alarm DB. Alarm DWs are numbered starting from 0, i.e. bytes 4-7 in the alarm DB will be the 0th alarm DW, bytes 8-11 the 1st alarm DW, etc.

AlmDW	Alarm DW contains 32 alarm bits. The FB will check whether some alarm bits are changed, compared with the previous alarm DW value. If there are no changes, the alarm DW will be ignored. If some bits are changed, the alarm DW will be copied to the alarm DW (specified by input parameter AlmDWNum) within alarm DB (specified by input parameter AlmDBNum) and respective bit timestamps in the alarm DB will be updated. If the CLServer has not yet acknowledged, that the previous alarm DW is read (by clearing MSB bit in the configuration DW), then the FB will store the received alarm DW and its timestamp in the alarm queue for later processing.
UseExtTS	"Use external timestamp" flag. If the value is FALSE, the FB will read the PLC clock, to obtain the internal timestamp for the alarm bits. If the value is TRUE, the FB will use an external timestamp, passed to the FB via the ExtTS parameter. The used timestamp (internal or external) will be passed back to the caller via the TS parameter.
ExtTS	External timestamp. The FB ALARMMNG will use this timestamp if the UseExtTS parameter value is TRUE.

Input/Output parameters

Init	"Initialize Alarm Manager" flag. If the value is TRUE, the FB will perform initialization, i.e. clear the alarm queues, etc. After initialization the FB will reset the Init parameter. Note, that on the first scan, the FB will perform initialization regardless of the Init parameter value.
------	--

Output parameters

The FB output parameters are not passed via the parameter list, but can accessed directly in the instance data block.

TS	Timestamp used for the changed alarm DW bits. If the alarm DW bits are not changed, compared with the previous FB call (and therefore were not saved and timestamped), the returned timestamp will have value DT#1990-01-01-00:00:00.000.
AlmQueLen	Current alarm queue length for the specified alarm DW. If a new alarm DW value is received, but the CLServer has not yet acknowledged, that the previous alarm DW is read (by clearing MSB bit in the configuration DW), then the FB will store the received alarm DW and its timestamp in the alarm queue. The AlmQueLen returns the current alarm queue length. The maximum alarm queue length, specified in the demo project, is 10 elements (element=alarm DW and timestamp). An alarm queue is supported separately for each alarm DW.
ErrCode	Error code: =0 – No error.

- =1 – Not an alarm DB. The specified alarm DB does not contain the constant 0xAA in its configuration DW (please, see the chapter “Configuration word specification”).
- =2 - Invalid number of alarm DWs configured. Check the number of alarm DWs, specified in the configuration DW. It should not be 0, and should not exceed constant AlmDWNuMax, defined in the FB ALARMMNG.
- =3 – Incompatible alarm DB. The AlmDBNum parameter value is not the same as on the first FB call. On the first FB call, the AlmDBNum is stored in the instance DB for the FB. The instance DB is used for alarm queues and must correspond to the alarm DB. The FB ALARMMNG can support multiple alarm DBs, but each alarm DB must have its own instance DB.
- =4 – invalid DW number. The AlmDWNuMax parameter value does not fit in the range defined in configuration DW for the alarm DB.
- =5 – Alarm queue overflow. In case of the alarm queue overflow, the oldest element (alarm DW and timestamp) will be deleted from the alarm queue. The alarm DW value will be lost and the timestamps, for the following alarm DWs in the queue, may be not updated correctly. (The alarm bit timestamp is updated if the bit value changes. If some alarm DW value is lost, the bit changes for the following alarm DWs cannot be tracked correctly).

Modifying array sizes

The FB ALARMMNG supports an alarm DB with 10 alarm DWs (320 alarm bits). To increase the number of alarms, there are two ways:

- Create a new alarm DB and call the FB ALARMMNG, indicating a new instance DB number (each alarm DB must have a linked instance DB).
- Modify the FB ALARMMNG to support a larger alarm DB. Below are the steps for modification:

1. Create new alarm DB and specify the number of supported alarm DWs in its configuration DW (please, see the chapter “Configuration word specification” above). Allocate memory for alarm DWs and the alarm bit timestamps.

2. Open the FB ALARMMNG and find the line, defining constant AlmDWNuMax:

```
AlmDWNuMax := 10; // MODIFY: Maximum number of alarm DWs
```

Modify the constant according to your requirements. The corresponding array sizes will be adjusted automatically.

If the CLServer has not yet acknowledged, that the current alarm data are processed (by clearing the MSB bit in the configuration DW), the FB ALARMMNG will store new alarm data in the queues (each alarm DW has its queue) for later passing to the CLServer. The default maximum alarm queue length is 10. To modify the maximum queue length, open the FB ALARMMNG and find the line, defining constant AlmQueMaxLen:

```
AlmQueMaxLen := 10; // MODIFY: Maximum alarm queue length
```

Modify the constant according to your requirements. The corresponding array sizes will be adjusted automatically.

Item names for CLServer

The “Demo example program for Siemens S7-300 PLC” supports the following I/O items range for CLServer:

DB20,X4.0 ... DB20, X43.7 (320 alarm bits in DB20)

DB30,X4.0 ... DB30, X43.7 (320 alarm bits in DB30)

DB40,X4.0 ... DB40, X43.7 (320 alarm bits in DB40)

Getting alarm timestamps from Allen-Bradley ControlLogix PLCs via DASABCIP

OPC Client part communication with DASABCIP and Allen-Bradley PLC

The CL Server/Client OPC Client part connects to the DASABCIP OPC server when some DDE/SuiteLink client connects to the configured CLServer topic, and disconnects from DASABCIP when all CLServer DDE/Suitelink clients are disconnected. The reading of data from DASABCIP is activated after activation any of data items and deactivated when all data items are deactivated and all pending OPC transactions are completed.

Note!

The alarm data in the Allen-Bradley PLC must be in double word (DINT) array format – it is the only data type supported by CLServer, i.e. all alarm information must be collected in alarm data array.

If OPC Client part receives new I/O item (alarm bit) activation request from SuiteLink/DDE Server part then following is done:

1) Checking if received I/O item(s) is valid – by checking item name syntax and reading and checking corresponding alarm data array configuration double word (dword), if not yet already checked by other I/O item of this alarm data array. The checking of configuration dword is done after the item activation, so if corresponding alarm data array does not exist or contains invalid data, then following happens:

- notification is logged to LogViewer,
- ERRORITEMS system item value is increased,
- item is marked as bad and have bad value quality.

No further activity of this item is possible except deletion.

If configuration dword content is successfully read and has correct values then CLServer activates OPC conversations with corresponding alarm data array. Contents of corresponding DB configuration dword indicates number of alarm dwords, alarm data array version number, alarm data array identification number and new alarms presence flag. One alarm dword can handle 32 alarm bits.

CLServer can receive item names pointing to alarm data arrays on several PLCs (topics).

2) If I/O item is valid, the continuous reading of MSB is started. MSB indicates if changes has been occurred in alarm words bit statuses (=some bit(s) has changed 0->1 or 1->0).

3) If alarm data array MSB (used by PLC for new alarm/event indication) is ON (non-zero) then depending on the configuration of CLServer topic:

a) if complete data block reading is configured then OPC Client part requests reading of full alarm data array, including: alarm double words = according the configuration word low word part and timestamps = according the amount of alarm double words;

b) if complete data block reading is not configured then OPC Client part requests reading of alarm double words = according the configuration word low word part; in this case timestamps are read ONLY for alarms with changed status.

Both for a) and b):

- all changed alarm bit and / or timestamp values and corresponding VTQ data is transferred to SuiteLink/DDE Server part and correspondingly to SuiteLink/DDE Client (InTouch);
- OPC Client resets the configuration word MSB and waits while OPC server will notify about MSB value change;

4) If alarm data array configuration word MSB is OFF (zero) then it means there are no new alarms/events (=changes) in this alarm data array, no special activity needed.

SuiteLink/DDE Server part

From SuiteLink/DDE Client (e.g. InTouch) point of view, the accessing to "CL Server/Client" SuiteLink/DDE Server part is completely same as accessing to DASABCIP, i.e. same topic names and I/O point naming syntax is used, the only difference is the Server application name is "CLServer".

For "CL Server/Client" SuiteLink/DDE Server part, the alarm bit I/O item names have the exactly same syntax as valid DASABCIP bit items, for example, DB0[1].5 specifies the bit 5 in double word 1 in alarm data array DB0.

CLServer offers additional (system) items to the SuiteLink/DDE clients, see *Monitoring and Controlling Communication with DAServers* section below in this manual.

ControlLogix PLC alarm data array layout

Configuration dword specification:

- configuration word is 32 bit long double word (bits 0...31)
- bit 31 = MSB used for change detection
- bit 30 = reserved for future use
- bit 29 = reserved for future use
- bit 28-24 of high word high byte are reserved for version control
 - o version number is between 1-31

- by using the version number, CLServer can detect the alarm data array structure (this possibility is reserved for future changes of alarm data array structure)
- high word low byte has a fixed value „AA” (in hex)
 - AA means that this alarm data array is used for alarms
 - This only confirms that this is really an alarm data array
- low word (bits 8..15) reserved for future use
- low word (bits 0..7) are used to inform amount of alarm double words; according to this information CLServer calculates the total space needed for alarm double words and timestamps

ControlLogix PLC alarm data array layout, version 1:

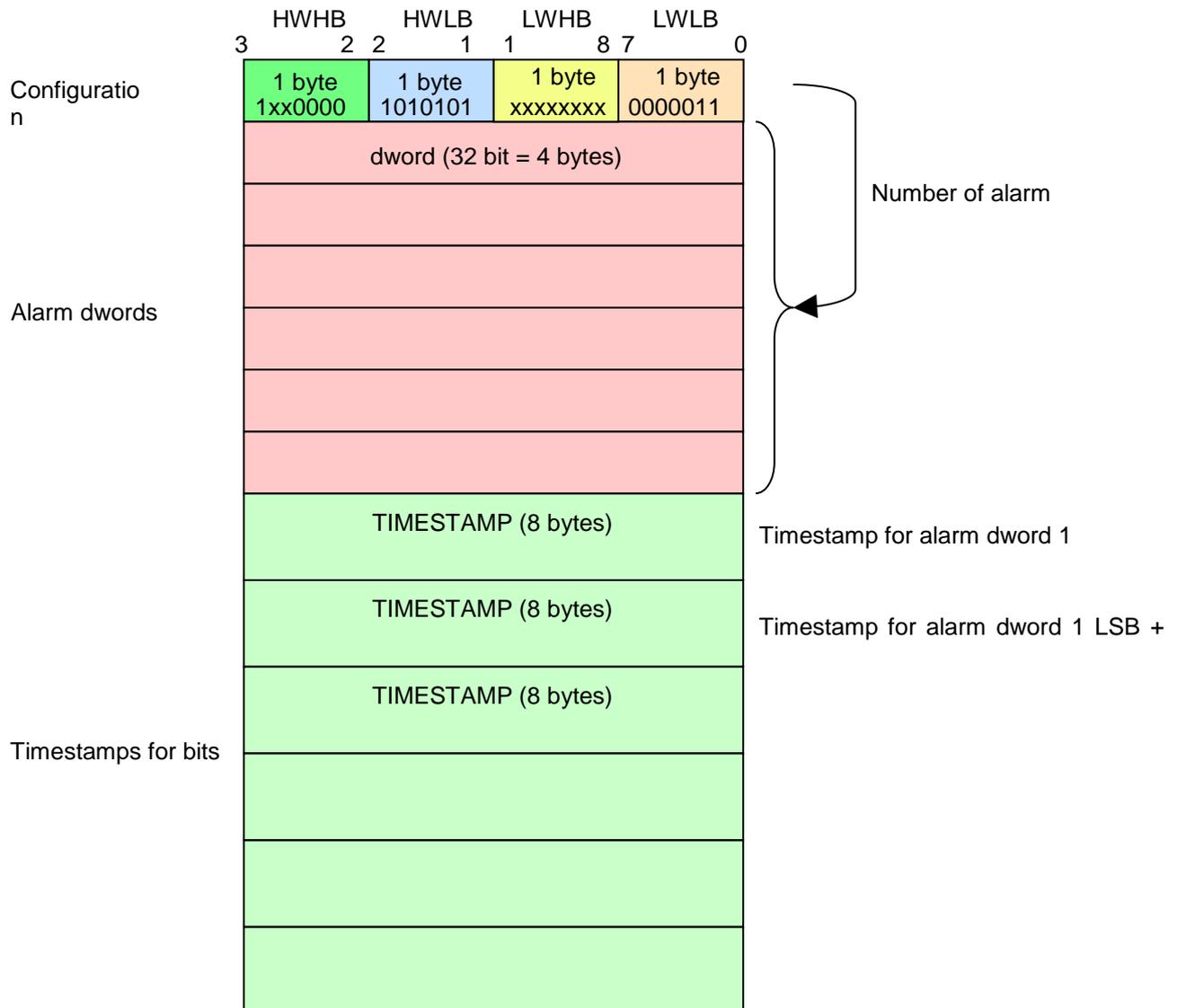
Configuration dword high word high byte (HWHB) MSB is used to indicate a change in the alarm dwords has occurred

Configuration dword high word high byte (HWHB) 5 least significant bits are used to indicate the version number of the alarm data array structure (range 1...31)

Configuration dword high word low byte (HWLB) contains the alarm data array identification number, hex AA (10101010)

Configuration dword low word high byte (LWHB) is not used (reserved for future use)

Configuration dword low word low byte (LWLB) is used to indicate the number of alarm dwords in the alarm data array



The size of the alarm data array (in bytes) is: $4 + LWLB * 4 + LWLB * 8 * 32$

Demo example program for ControlLogix PLC

CL Server/Client distribution package contains the "Demo example program for Allen-Bradley ControlLogix PLC" (**CLogix_Demo.zip**). This demo program can be used as an example how to add appropriate additions/modifications in Allen-Bradley ControlLogix PLC program from where the alarm/event timestamping data is coming.

The ControlLogix PLC has to support the alarm data array structure, described in the chapter "Configuration word specification". Therefore, an RSLogix 5000 demo project for ControlLogix PLC is provided (please, see the file ControlLogix_AlarmMng.ACD), containing the routine ALARMMNG. The routine ALARMMNG handles the alarm data array DB0 and alarm queues.

The demo project has a following call trees:

```
AlarmGenTask
  AlarmGenProgram
    ALARMGEN
```

```
MainTask
  MainProgram
    MainRoutine
      ALARMMNG
      S_TOD
```

The alarms, for demonstration purposes, are generated by the routine ALARMGEN, called from the AlarmGenProgram every 10 ms. The ALARMGEN routine changes alarm bits by shifting left the contents of alarm double words (32 alarm bits each) at the specified time intervals. The ALARMGEN routine provides 10 alarm double words. The default time intervals for them are: 300, 120, 60, 30, 20, 10, 5, 3, 2, 1 seconds.

The routine ALARMMNG is called on every PLC scan cycle. The routine call has a following format:

```
JSR (ALARMMNG,          // Alarm Manager routine
     5,                 // Number of input parameters
     MR_AlmDWNNum,     // Alarm double word number
     MR_AlmDW,         // Alarm double word (32 alarm bits)
     MR_UseExtTS,      // "Use external timestamp" flag
     MR_ExtTS,         // External timestamp
     MR_Init,          // "Initialize Alarm Manager" flag
     MR_TS,            // Timestamp
     MR_AlmQueLen,    // Alarm queue length
     MR_ErrCode);     // Error code
```

Global parameters

DB0 Alarm data array. The alarm data array is not passed to the ALARMMNG routine via the input/output parameters, but is defined as a global (controller) tag and accessed directly. Alarm data array contains the configuration double word, alarm double words (32

alarm bits each) and alarm bit timestamps, as described in the chapter "Configuration word specification".

Input parameters

MR_AlmDWNuM	Alarm double word (DW) number within the alarm data array DB0. Note, that it is not an array index. DB0[0] will contain configuration DW. DB0[1] will contain the 0 th alarm DW, DB0[2] the 1 st alarm DW, etc.
MR_AlmDW	Alarm DW contains 32 alarm bits. The ALARMMNG routine will check whether some alarm bits are changed, compared with the previous alarm DW value. If there are no changes, the alarm DW will be ignored. If some bits are changed, the alarm DW will be copied to the alarm DW (specified by input parameter MR_AlmDWNuM) within alarm data array and respective bit timestamps in the alarm data array DB0 will be updated. If the CLServer has not yet acknowledged, that the previous alarm DW is read (by clearing MSB bit in the configuration DW), then the routine will store the received alarm DW and its timestamp in the alarm queue for later processing.
MR_UseExtTS	"Use external timestamp" flag. If the value is FALSE, the routine will read the PLC clock, to obtain the internal timestamp for the alarm bits. If the value is TRUE, the routine will use an external timestamp, passed to the routine via the MR_ExtTS parameter. The used timestamp (internal or external) will be passed back to the caller via the MR_TS parameter.
MR_ExtTS	External timestamp. The ALARMMNG routine will use this timestamp if the MR_UseExtTS parameter value is TRUE.
MR_Init	"Initialize Alarm Manager" flag. If the value is TRUE, the routine will perform initialization, i.e. clear the alarm queues, etc. Note, that on the first scan, the routine will perform initialization regardless of the MR_Init parameter value.

Output parameters

MR_TS	Timestamp used for the changed alarm DW bits. If the alarm DW bits are not changed, compared with the previous routine call (and therefore were not saved and timestamped), the returned timestamp will have 0's in all fields (year, month, ...).
MR_AlmQueLen	Current alarm queue length for the specified alarm DW. If a new alarm DW value is received, but the CLServer has not yet acknowledged, that the previous alarm DW is read (by clearing MSB bit in the configuration DW), then the routine will store the received alarm DW and its timestamp in the alarm queue. The MR_AlmQueLen returns the current alarm queue length. The maximum alarm queue length, specified in the demo project, is 10

elements (element=alarm DW and timestamp). An alarm queue is supported separately for each alarm DW.

MR_ErrCode	<p>Error code.</p> <p>=0 – No error.</p> <p>=1 – Not an alarm data array. The array does not contain the constant 0xAA in its configuration DW (please, see the chapter “Configuration word specification”).</p> <p>=2 - Invalid number of alarm DWs configured. Check the number of alarm DWs, specified in the configuration DW, and the alarm data array DB0 size. See the formula for DB0 array size calculation at the end of this chapter.</p> <p>=3 – invalid alarm DW number. The MR_AlmdWNum parameter value does not fit in the range defined in configuration DW for the alarm data array.</p> <p>=4 – invalid AlmQue array size. See the formula for AlmQue array size calculation at the end of this chapter.</p> <p>=5 – invalid AlmdWChn array size. See the formula for AlmdWChn array size calculation at the end of this chapter.</p> <p>=6 – Alarm queue overflow. In case of the alarm queue overflow, the oldest element (alarm DW and timestamp) will be deleted from the alarm queue. The alarm DW value will be lost and the timestamps, for the following alarm DWs in the queue, may be not updated correctly. (The alarm bit timestamp is updated if the bit value changes. If some alarm DW value is lost, the bit changes for the following alarm DWs cannot be tracked correctly).</p>
------------	--

Modifying array sizes

The routine ALARMMNG supports an alarm data array DB0 with 10 alarm DWs (320 alarm bits). To change the number of supported alarms, modify the size of the DB0, AlmQue and AlmdWChn arrays. The array sizes can be calculated using the following formulas:

$$\text{DB0 size} = 1 + \text{Number of alarm DWs} + \text{Number of alarm DWs} * 64$$

$$\text{AlmQue size} = \text{Number of alarm DWs} * (3 * (\text{Maximum alarm queue length} + 1) + 1)$$

$$\text{AlmdWChn size} = \text{Number of alarm DWs} / 32 + 1$$

If the number of alarm DWs is a multiple of 32, then the AlmdWChn array size can be calculated using formula:

$$\text{AlmdWChn size} = \text{Number of alarm DWs} / 32$$

Note, that:

Number of alarm DWs > 0;

Maximum alarm queue length > 0;

If the new alarm data cannot be immediately passed to CLServer for processing, because the CLServer has not yet acknowledged the processing of the previous alarm data, the

routine ALARMMNG will store new alarm data in the queue. The default maximum alarm queue length is 10. To modify the maximum alarm queue length, modify the AlmQue array size, according to the formula specified above.

Item names for CLServer

The “Demo example program for Allen-Bradley ControlLogix PLC” supports the following I/O items range for CLServer:

DB0[1].0 ... DB0[10].31 (320 alarm bits in array “DB0”)

Getting alarm timestamps from Modicon PLCs via DASMBTCP

OPC Client part communication with DASMBTCP and Modicon PLC

The CL Server/Client OPC Client part connects to the DASMBTCP OPC server when some DDE/SuiteLink client connects to the configured CLServer topic, and disconnects from DASMBTCP when all CLServer DDE/SuiteLink clients are disconnected. The reading of data from DASMBTCP is activated after activation any of data items and deactivated when all data items are deactivated and all pending OPC transactions are completed.

Note!

The alarm data in the Modicon PLC must be grouped in alarm area – some consecutive Holding Registers area and all alarm information must be collected into this alarm area by Modicon PLC program. This alarm area start address is configured in CLServer topic linked with this Modicon PLC.

If OPC Client part receives new I/O item (alarm bit) activation request from SuiteLink/DDE Server part then following is done:

1) Checking if received I/O item(s) is valid – by checking item name syntax and reading and checking corresponding alarm area configuration double word (alarm area first two registers), if not yet already checked by other I/O item of this alarm area. The checking of configuration dword is done after the item activation, so if corresponding alarm area does not exist or contains invalid data, then following happens:

- notification is logged to LogViewer,
- ERRORITEMS system item value is increased,
- item is marked as bad and have bad value quality.

No further activity of this item is possible except deletion.

If configuration dword content is successfully read and has correct values then CLServer activates OPC conversations with corresponding alarm area. Contents of corresponding alarm area configuration dword indicates the total number of alarm dwords (2 consecutive registers each, 32-bit totally in one dword) used, alarm area version number, alarm area identification number and new alarms presence flag. One alarm dword handles 32 alarm bits.

CLServer can receive item names pointing to alarm areas on several PLCs (topics).

- 2) If I/O item is valid, the continuous reading of alarm area configuration dword MSB is started. MSB indicates if changes has been occurred in alarm bits statuses (=some bit(s) has changed 0->1 or 1->0).
- 3) If alarm area configuration dword MSB (used by PLC for new alarm/event indication) is ON (non-zero) then depending on the configuration of CLServer topic:
- a) if complete alarm area reading is configured then OPC Client part requests reading of full alarm area, including: alarm dwords = according the configuration dword and timestamps = according the amount of alarm dwords;
- b) if complete alarm area reading is not configured then OPC Client part requests reading of alarm dwords = according the configuration dword ; in this case timestamps are read ONLY for alarms with changed status.
- Both for a) and b):
- all changed alarm bit and / or timestamp values and corresponding VTQ data is transferred to SuiteLink/DDE Server part and correspondingly to SuiteLink/DDE Client (InTouch);
 - OPC Client resets the configuration dword MSB and waits while OPC server will notify about MSB value change;
- 4) If alarm data area configuration dword MSB is OFF (zero) then it means there are no new alarms/events (=changes) in this alarm area, no special activity needed.

SuiteLink/DDE Server part

From SuiteLink/DDE Client (e.g. InTouch) point of view, the accessing to "CL Server/Client" SuiteLink/DDE Server part is completely same as accessing to DASMBTCP, i.e. same topic names and I/O point naming syntax is used, the only difference is the Server application name is "CLServer".

For "CL Server/Client" SuiteLink/DDE Server part, the alarm bit I/O item names have the exactly same syntax as valid DASMBTCP "bit in register" items, for example, 401278:12 specifies the bit 12 in Holding Register 401278.

CLServer offers following additional (system) items to the SuiteLink/DDE clients:

1. STATUS – discrete, indicates OPC communication status with DASMBTCP. The status value 1 – means OK , 0 – FAILED.
2. ITEMCOUNT – integer, indicates number of active SuiteLink/DDE items including system items and ITEMCOUNT itself.
3. ERRORCOUNT – integer, indicates number of OPC errors, received bad quality values occurred since topic initialization or last reset of errorcount. ERRORCOUNT value is reseted after writing 0 to item ERRORCOUNT.
4. ERRORITEMS – integer, indicates number of invalid items which refers to invalid datablock.
5. WATCHDOG – integer, indicates number of seconds since topic activation.
6. READTIME – integer, indicates number of milliseconds spent to read alarm data.
7. WRITETIME – integer, indicates number of milliseconds between writing MSB and writing status confirmation.

Note:

For item naming of alarm bits the bit position inside register can be between 1 and 16 or from 16 to 1 (most significant bit can be 1 or 16) – this bit order is defined in DASMBTCP TSXQuantum or TSXPremium object configuration. CLServer detects the currently used bit order by reading special bit in Modicon PLC alarm area (see *Modicon PLC alarm area specification* section below).

Modicon PLC alarm area specification

The alarm data in the Modicon PLC must be grouped in alarm area – some consecutive Holding Registers area and all alarm information must be collected into this alarm area by Modicon PLC program. This alarm area start address is configured in CLServer topic linked with this Modicon PLC.

This alarm area should have the following contents:

Holding Register address in alarm area	Description
“Alarm area” start address	Header High Register: low byte is used to inform the number of currently used alarm dwords (N) - according to this information CLServer calculates the total space needed for alarm dwords and timestamps; high byte is reserved for future use
“Alarm area” start address + 1	Header Low Register: low byte contains a fixed value „AA” (in hex) indicating that this is really an alarm area; 3 high byte bits (MSBs) are used for alarm states change detection, bit 4 is used for bit order detection (bit value must be 0), bits 5...8 are reserved for version control (version number between 1-15)
“Alarm area” start address + 2	Low word of 1 st alarm dword
“Alarm area” start address + 3	High word of 1 st alarm dword
“Alarm area” start address + 4	(N – 1) next alarm dwords
...	
“Alarm area” start address + 2 + N x 2	Timestamp of first alarm dword low bit
“Alarm area” start address + 2 + N x 2 + 5	Timestamp of first alarm dword low bit + 1
...	N x 32 timestamps (5 registers for each alarm)

So, the total amount of space required for alarm area can be calculated the following way (N is the the number of currently used alarm dwords):

In dwords: **1 + N + N x 80**

In registers: **2 + N x 2 + N x 160**

For example, if N = 10 (10 alarm dwords = 320 alarm bits) then space required for alarm area is 811 dwords (or 1622 registers).

The timestamp for each alarm is stored in 5 consecutive registers, in following sequence:

1st (least significant) register – “unused” byte and “seconds” byte
 2nd register – “minutes” byte and “hours” byte
 3rd register – “day” byte and “month” byte
 4th register – year
 5th (most significant) register – milliseconds
 All timestamp fields are in BCD format.

Notes:

1. In “CLServer Topic Definition” the **MSB Offset** setting is used to specify which bit of alarm area configuration dword is used as MSB. If MSB Offset is 0 then 1st bit is used as MSB, if it is 1 then 2nd bit is used as MSB and if it is 2 then 3rd bit is used as MSB for this topic - all in case the MSB offset is not specified by CLServer program startup parameters.
2. Bit order detection is used only for DASMBTCP server. So, the meaning of 4th bit in configuration DWORD “Alarm area” start address + 1 high byte for other OPC Servers is uncertain. Anyway, the existing version of CLServer must reject all data block with version higher than 1.

Getting alarm timestamps from Omron PLCs via DASOmronEth

OPC Client part communication with DASOmronEth and Omron PLC
 The CL Server/Client OPC Client part connects to the DASOmronEth OPC server when some DDE/SuiteLink client connects to the configured CLServer topic, and disconnects from DASOmronEth when all CLServer DDE/Suitelink clients are disconnected. The reading of data from DASOmronEth is activated after activation any of data items and deactivated when all data items are deactivated and all pending OPC transactions are completed.

Note!

The alarm data in the Omron PLC must be collected in consecutive “alarms area”, located in Data Memory area. The minimum size of such “alarms area” is 6 words – in this case 16 alarm bits are supported.

The “alarms area” has the following contents:

“Alarms area” in Omron PLC Data Memory Area	Description
“Alarms area” start address	Header, contains the version number and number of alarm data words currently used
“Alarms area” start address + 1	Alarm counter – increases at each alarm bit change
“Alarms area” start address + 2	Seconds (00-07 bits) & Minutes (08-15 bits)
“Alarms area” start address + 3	Hours (00-07 bits) & Date (08-15 bits)
“Alarms area” start address + 4	Month (00-07 bits) & Year (08-15 bits)
“Alarms area” start address + 5	Alarm data 1 st word, includes 16 alarm bits
...	
“Alarms area” start address + n	Alarm data last word, includes 16 alarm bits

From above table you can see - the timestamping data must occupy 3 consecutive words in DM Area (starting from "Alarms area start address + 2"), where at each PLC cycle the PLC Auxiliary Area Clock data (from A351 to A353) is copied inside the PLC program.

Unlike in Siemens S7 or Allen-Bradley ControlLogix PLCs, in Omron PLCs the PLC clock does not support milliseconds – one second is the shortest time unit available. For this reason, the CLServer support for alarm/event timestamps from Omron PLCs is much more simple as for S7 or Allen-Bradley ControlLogix PLCs - there is no buffering of alarms and special new alarm/event indication bit used: CLServer simply reads (via DASOmronEth) the "alarms counter" from PLC and if "alarms counter" value has changed (indicating some alarm bit has changed its state) then whole "alarms area" is immediately read at once and all changed alarm bit values and current timestamp as VTQ data are provided to SuiteLink/DDE Clients.

So, if OPC Client part receives new I/O item (alarm bit) activation request from SuiteLink/DDE Server part then following is done:

Checking if received I/O item(s) is valid – by checking item name syntax and reading and checking corresponding "alarms area" contents from PLC (in case this is a first item activated for PLC); if requested item is not valid or is located outside "alarm area", then following happens:

- notification is logged to LogViewer,
- ERRORITEMS system item value is increased,
- item is marked as bad and have bad value quality.

No further activity of this item is possible except deletion.

If "alarms area" has correct values and requested I/O item really is some alarm bit from "alarms area" then CLServer starts the continuous reading of "alarms area" alarm counter (located in "alarms area" start address + 1) – the update rate depends on Update Interval configured for corresponding Device Group in DASOmronEth Server CONTROLLER object configuration (for Device Groups see the description of "Item Path" in "*Topic Definition Command*" chapter later in this manual); the value 200..300 milliseconds of DASOmronEth Device Group Update Interval would be far enough to obtain proper timestamps for alarms/events from Omron PLCs. So, in case also other clients (e.g InTouch directly) are communicating with DasOmroneth then it is recommended create special Device Group for CLServer with suitable Update Rate.

All changed alarm bit and / or timestamp values and corresponding VTQ data is transferred to SuiteLink/DDE Server part and correspondingly to SuiteLink/DDE Client (InTouch)

SuiteLink/DDE Server part

From SuiteLink/DDE Client (e.g. InTouch) point of view, the accessing to CL Server/Client" SuiteLink/DDE Server part is completely same as accessing to DASOmronEth, i.e. same topic names and I/O point naming syntax is used, the only difference is the Server application name is "CLServer".

For "CL Server/Client" SuiteLink/DDE Server part, the alarm bit I/O item names have the exactly same syntax as valid DASOmronEth bit items, for example, D1005.5 specifies the bit 5 in Data Memory area word 1005 (this is 6th alarm bit in "alarms area").

CLServer offers additional (system) items to the SuiteLink/DDE clients, see *Monitoring and Controlling Communication with DAServers* section below in this manual.

Demo example program for Omron CS1 PLC

CL Server/Client distribution package contains the "Demo example program for Omron CS1 PLC" (**CS1_Demo.zip**). This demo program can be used as an example how to add appropriate additions/modifications in Omron CS1 PLC program from where the alarm/event timestamping data is coming.

In this demo example program, the "alarms area" starts from D1000 and occupies 6 consecutive words:

"Alarms area" in Omron PLC Data Memory Area	Description and contents
D1000	Header, contains 1A01, where 1 – the version number, A – delimiter and 01 – the number of alarm data words
D1001	Alarm counter – increases at each new alarm
D1002	Seconds (00-07 bits) & Minutes (08-15 bits)
D1003	Hours (00-07 bits) & Date (08-15 bits)
D1004	Month (00-07 bits) & Year (08-15 bits)
D1005	Alarm data word, includes 16 alarm bits

Most important in this CS1 demo example program:

1) At PLC start-up:

- value #1 is moved to CIO address 10 – for use in alarm generation by shifting bits left at each Timer1 cycle,
- initial value #0 is moved to D1001 ("alarm counter"),
- value #1A01 moved to D1000 ("alarms area" header),
- value #10 (units of 0.1 seconds) is moved to D999, which is used as a set value for Timer1.

2) At each Timer1 cycle:

- contents of CIO address 10 is shifted by one position left and copied to D1005, so new alarm is generated;
- "alarm counter" D1001 is increased by 1;
- contents of PLC clock (A351...A353) is copied to D1002...D1004.

Item names for CLServer

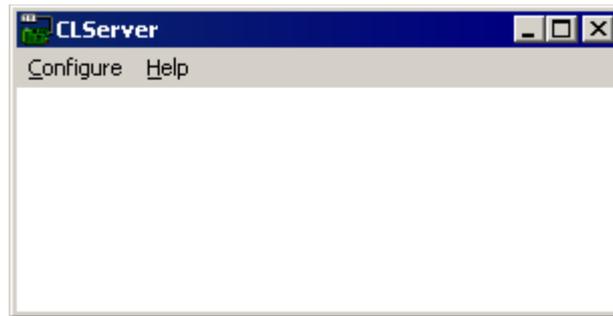
The "Demo example program for Omron CS1 PLC" supports the following I/O items range for CLServer:

D1005.0 ... D1005.15 (16 alarm bits)

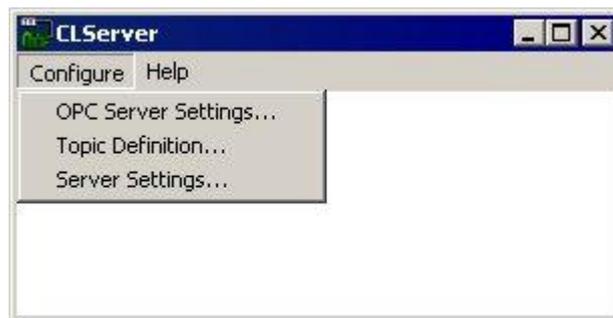
Configuring the CLSERVER

After the CLServer is initially installed, a little of configuration is required. Configuring the Server automatically creates **CLSERVER.CFG** file that holds all of the OPC Server settings and topics definitions entered. This file will be placed automatically in the same directory in which the CLServer is located unless the path where the configuration file will be placed is specified via the */Configure/Server Settings...* command.

To perform the required configurations, start up the CLServer program. If the CLServer starts up as an icon, double-click on the icon to open the server's window. The following will appear:



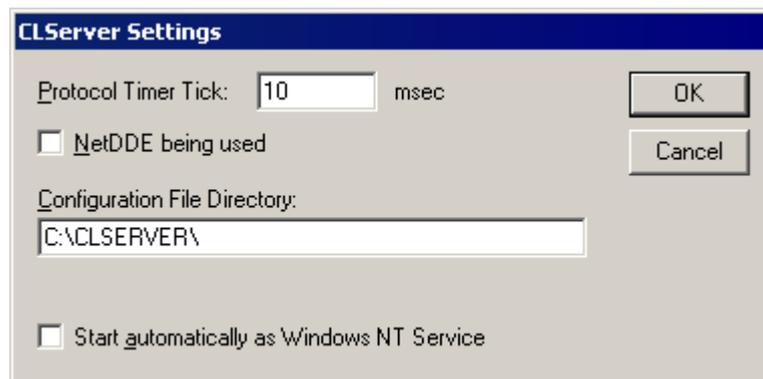
To access the commands used for the various configurations, open the */Configure* menu:



Server Settings Command

A number of parameters that control the internal operation of the Server can be set. In most cases, the default settings for these parameters provide good performance and do not require changing. However, they can be changed to fine-tune the Server for a specific environment.

To change the Server's internal parameters, invoke the */Configure/Server Settings...* command. The "CLServer Settings" dialog box will appear:



The following describes each field in this dialog box:

Protocol Timer Tick

This field is used to change the frequency at which the CLServer checks for work to do - at this frequency CLServer checks if there are new data obtained from corresponding DAServers and delivers new data to SuiteLink or DDE Client(s). If computer is very busy or some other MS Windows application is taking over the computer then the Server is activated rarely than setting in the **Protocol Timer Tick**.

Note: *The default value is 50 milliseconds. The minimum value is 10 milliseconds.*

NetDDE being used

Select this option if you are networking using NetDDE.

Configuration File Directory

The first field is used to specify the path (disk drive and directory) in which CLServer will save its current configuration file. CLServer will use this path to load the configuration file the next time it is started.

Note: *Only the "path" may be modified with this field. The configuration file is always named **CLSERVER.CFG**.*

Note: *There is no limit to the number of configuration files created, although each must be in a separate directory. When using the CLServer with **InTouch**, it is good practice to place the configuration file in the application directory.*

Start automatically as Windows NT Service

Enabling this option will cause the server to start as a Windows NT service.

Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause the Server to run as a Windows NT service. However, to view configuration information or to reconfigure the Server, the user must log on to the system. Any Server related problems that may arise such as missing adapter cards, licensing failures or device drivers not loading will not be visible to the user until a log on is performed. Disabling this option and rebooting the system will cause the Server to run as a Windows NT application program once again.

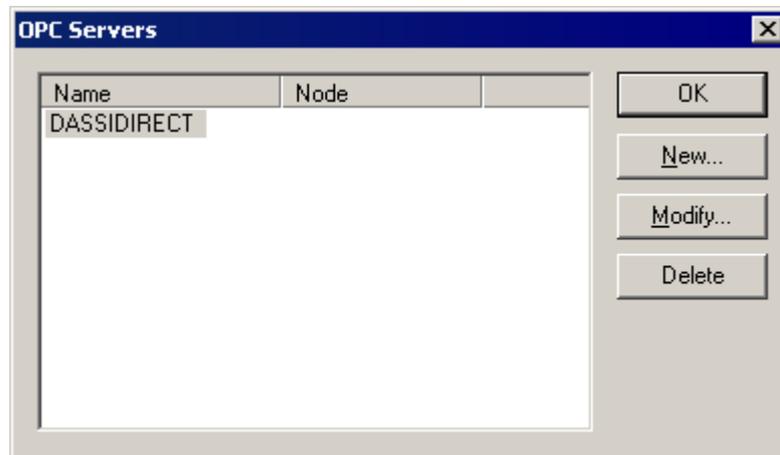
Note:

*The Service Startup configuration can be changed by MS Windows NT **Control Panel/Services** configuration dialogs. The **Allow Service to Interact with Desktop** checkbox in "Service" dialog box must be checked (the "Service" dialog box can be invoked by pressing the "Startup" button on "Services" dialog box when Service **CLSERVER_IOServer** is selected). If **Allow Service to Interact with Desktop** is not selected then CLServer full functionality is not ensured (e.g. the Server configuration can not be changed, no message boxes will be displayed, etc.).*

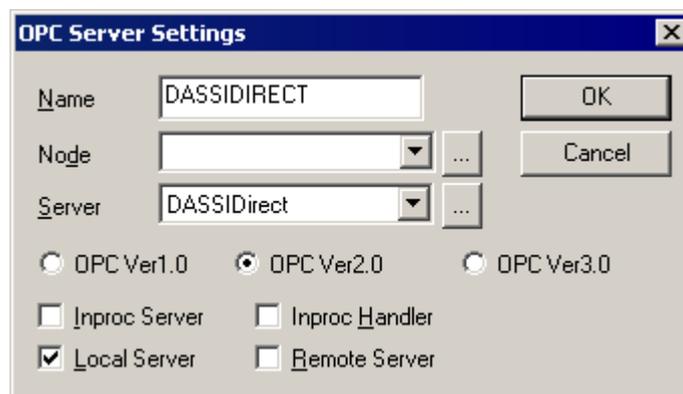
Once all entries have been made, click on **OK**.

OPC Server Settings Command

This command is used to select and configure the CL Server/Client OPC Client part - the settings for OPC Server (corresponding DASServer), used for communication with PLCs and to which the CL Server/Client OPC Client part will connect and communicate. Invoke the `/OPC Server Settings...` command. The "OPC Servers" dialog box will appear:



Click on **New** to create a new OPC Server definition or select the already defined OPC Server and click on **Modify** to edit the OPC Server settings - the "OPC Server Settings" dialog box will appear (the settings below are default ones supplied with default configuration file CLSERVER.CFG):



The following can be entered in this dialog box:

Name

Enter the name of OPC Server. Any name can be entered here, this name will be used in "Topic definition" dialog as a reference to OPC Server.

Node

Select the Node (computer name in local network) where the OPC Server resides.

Server

Select (browse) the OPC Server (currently the supported ones are Wonderware DASSIDirect, DASABCIP, DASMBTCP and Klinkmann's DASOmronEth) from the list of available OPC Servers installed and registered on the selected Node.

OPC standard version

Select the OPC standard (1.0, 2.0 or 3.0) supported by OPC Server. The default setting for is OPC Ver2.0.

OPC connection type

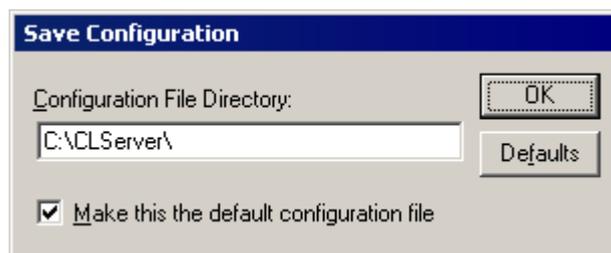
Select the OPC connection type(s) supported by OPC Server. The default and recommended setting is "Local Server".

Click **OK** to accept the new OPC Server settings and close the "OPC Server Settings" dialog box or click **Cancel** to reject the entered settings. The "OPC Server" dialog box will reappear.

Click **OK** to close the "OPC Server" dialog box.

Saving of the CLSERVER Configuration File

If the configuration file does not currently exist, or a new configuration path has been specified, the Server will display the "Save Configuration" dialog box:



This dialog box displays the path where the CLServer is going to save the current configuration file. The path may be changed if necessary. Also, the path can optionally be recorded in the **WIN.INI** file by selecting the "**M**ake this the default configuration file" option. Doing so will allow the CLServer to find the configuration file automatically each time it is started.

Configuration File Location

When the CLServer starts up, it first attempts to locate its configuration file by, first checking the **WIN.INI** file for a path that was previously specified. If the path is not present in the **WIN.INI** file, the Server will assume that the current working directory is to be used.

To start the Server from an application directory configuration file other than the default configuration file a special switch (**/d:**) is used. For example, invoke the **File/Run** command and enter the following:

CLSERVER /d:c:\directoryname

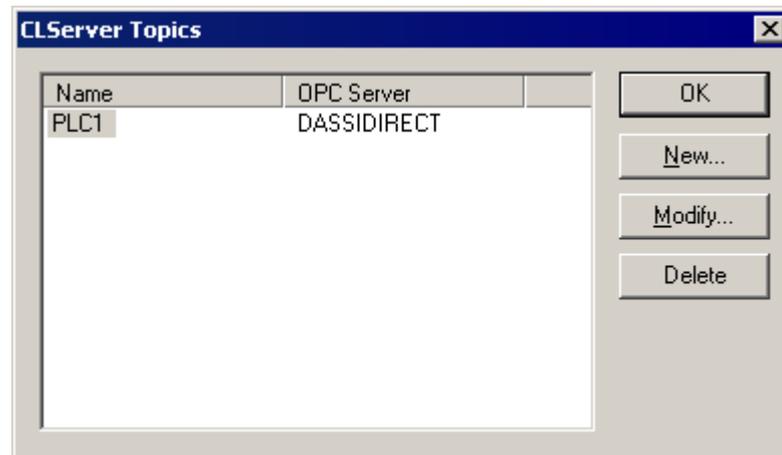
Note: There is no limit to the number of configuration files that may be created, although each must be in a separate directory.

Topic Definition Command

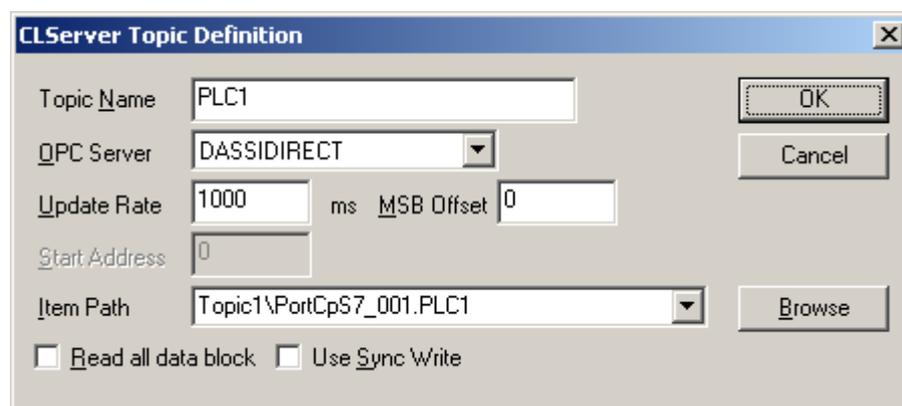
The user provides each PLC with an arbitrary name that is used as the Topic Name for all references to the node.

The following steps are taken to define the topics (controllers) connected to CLServer:

1. Invoke the Configure/Topic Definition... command. The "CLServer Topics" dialog box will appear:



2. To modify an existing topic, select the topic name and click on **Modify**. To define a new topic, click on **New**. The "CLServer Topic Definition" dialog box will appear:



3. Enter the **Topic Name**. If using **InTouch** the same Topic Name is to be entered in the "Add Access Name" dialog box described in the *Using the CLServer with InTouch* section.
4. Click on the **OPC Server** button to associate a topic with the OPC Server (only OPC Server names entered in "OPC Server Settings" dialog box are available here). Additional topics may be associated with the same OPC Server later.
5. Set the **Update Rate** field to specify the "update rate" for CLServer internally created OPC Group used to access the corresponding DAServer items for this Topic. This is a time how often OPC Server updates its cache and sends notifications to OPC Client (CLServer) in purpose to keep the data fresh (the OPC Server should never send data to OPC Client at a rate faster than the client requests). Default value **0** means the data are updated as fast as possible, some non-zero value can be sometimes useful to decrease the load of OPC Server.

6. **MSB Offset** specifies which bit of alarm data block configuration Dword is used. For **DASSIDIRECT** and **DASABCIP**, if MSB Offset is 0 (default) then 31st bit is used as MSB, if it is 1 then 30th bit is used as MSB and if it is 2 then 29th bit is used as MSB for this topic. For **DASMBTCP**, if MSB Offset is 0 (default) then “Alarm area” start address + 1 high register ‘highest bit’ is used as MSB, if it is 1 then ‘highest bit - 1’ is used as MSB and if it is 2 then ‘highest bit - 2’ is used as MSB for this topic. It is also possible to specify the MSB offset by program startup parameters, see *Program startup parameters* section below – in this case the startup parameter has higher priority as value specified in Topic definition.
7. For **DASMBTCP** topic, enter the **Start Address** of Modicon PLC alarm area, e.g. 412500.
8. Enter (browse) the **Item Path** for accessing the OPC Server items. The Access Path can be selected by browsing (clicking on Browse button, recommended) or can be entered manually.

For **DASSIDIRECT**, the Access Path is in the following format:

DeviceGroup/PortCpS7Object Name.S7CpObject Name

where **DeviceGroup**, **PortCpS7Object** and **S7CpObject** are corresponding objects in **DASSIDIRECT** hierarchy, configured in ArcestrA System Management Console (SMC).

For **DASABCIP**, the Access Path is in the following format:

DeviceGroup/PORT_CIP.ENB_CLX.BACKPLANE_CLX.LOGIX5000_CLX

For **DASMBTCP**, the Access Path is in the following format:

DeviceGroup/TCPIP_PORTObject Name.TSXQuantumObject Name
or
DeviceGroup/TCPIP_PORTObject Name.TSXPremiumObject Name

where **DeviceGroup**, **TCPIP_PORTObject**, **TSXQuantumObject** and **TSXPremiumObject** are corresponding objects in **DASMBTCP** hierarchy, configured in SMC.

For **DASOmronEth**, the Access Path is in the following format:

DeviceGroup/SOCKET.CONTROLLER

where **DeviceGroup**, **PortCpS7**, **S7CpObject**, **PORT_CIP**, **ENB_CLX**, **BACKPLANE_CLX**, **LOGIX5000_CLX**, **SOCKET** and **CONTROLLER** are corresponding objects in **DAServers** hierarchy, configured in ArcestrA System Management Console (SMC).

9. **Read all data block** checkbox can be used to force the reading of all alarm bit timestamps each time when MSB indicates new alarms. If unchecked then CLServer

reads timestamps only for changed alarms. Not used if DASOmronEth selected as "OPC Server".

10. **Use Sync Write** checkbox can be used to specify how to reset MSB when all alarm bits and timestamps are read. If checked then synchronous OPC writes are used, and if not checked then asynchronous OPC writes are used. The default setting is "unchecked". Not used if DASOmronEth selected as "OPC Server".

Once all entries have been made, click on **OK**.

Select **OK** in "Topic definition" dialog box when configuration for all Topics has been performed.

Item Names

The CLServer supports only the following items/points: alarm bits, alarm bit timestamps and some special items for monitoring and controlling the communication.

Item Names for DASSIDIRECT and DASMBTCP

The **alarm bits** are bit type items (I/O Discrete for InTouch) and provide VTQ data for VTQ-aware clients: alarm state (0 or 1), timestamp from S7 or Modicon PLC when alarm appears/disappears and item quality. The CLServer item/point naming for alarm bits completely matches with item/point naming used in DASSIDIRECT for **Bit** type items and in DASMBTCP for **Bit in Register** type items, in general the syntax is following:

For S7 PLCs:

D<B,I>d,Xx.y

where:

- d** - the data block number, with a range from 1 to 65535;
- x** - the start address of byte in data block, with a valid range from 4 to <3 + number of alarm double words * 4> (bytes 0...3 are occupied by alarm DB configuration word);
- y** - the bit position, with a range from 0 to 7, where 0 is the least significant bit and 7 is the most significant bit.

For Modicon PLCs:

r:y

where:

- r** - the Holding register address, with a range from 400001 to 465536;
- y** - the bit position inside register, with a range from 1 to 16; the bit order 1...16 or 16...1 (most significant bit can be 1 or 16) is defined in DASMBTCP TSXQuantum or TSXPremium object configuration.

The **alarm bit timestamps** are string type items (I/O Message type for InTouch), which can be used additionally to alarm bits; the string will contain the timestamp from S7 PLC

when alarm appears/disappears in the following range: from “1990-1-1-0:00:00.000” to “2089-12-31-23:59:59.999” – this format is completely compatible with DASSIDIRECT Byte and Char type items with suffix DT or with DASMBTCP String (ASCII) type items. The item/point name for alarm bit timestamp item needs the suffix **DT** to be added to alarm bit item name:

D<B,I>d,Xx.yDT

or

r:yDT

For alarm bit timestamp items, the VTQ timestamp will be the CLServer computer time when string value was provided for client, not S7 or Modicon PLC actual timestamp.

Note!

Please consider the following – there is a Big-Endian byte order (in words in double-words) used in Siemens PLCs. For example, in the 1st alarm double-word the byte order is following (from low byte to high byte): byte 7, byte 6, byte 5, byte 4; correspondingly the alarm timestamp data are located in the following order:

- alarm timestamp for byte 7 bit 0 (item name DBd,X7.0)
- alarm timestamp for byte 7 bit 1 (item name DBd,X7.1)
- ...
- alarm timestamp for byte 7 bit 7 (item name DBd,X7.7)
- alarm timestamp for byte 6 bit 0 (item name DBd,X6.0)
- ...
- alarm timestamp for byte 4 bit 7 (item name DBd,X4.7)

Above sequence must be supported in Siemens PLC program where alarm timestamps processing is enabled.

Item/point naming examples for S7 PLCs:

DB123,X4.0 – I/O Discrete type item, data block 123, bit 0 in byte 4; this is 1st alarm bit in alarm DB123

DB123,X4.5 – I/O Discrete type item, data block 123, bit 5 in byte 4; this is 6th alarm bit in alarm DB123

DI5,X9.0 – I/O Discrete type item, data block 5, bit 0 in byte 9; this is 41st alarm bit (32+8+1) in alarm DB5

DI5,X9.0DT – I/O Message type item, data block 5, bit 0 in byte 9; this is a timestamp for 41st alarm bit (32+8+1) in alarm DB5

Item/point naming examples for Modicon PLCs:

412502:2 – I/O Discrete type item, 2nd bit in Holding Register 412502; this is Modicon PLC 2nd alarm bit in case alarm area starts from address 412500

412502:2DT – I/O Message type item, 2nd bit in Holding Register 412502; this is a timestamp for Modicon PLC 2nd alarm bit in case alarm area starts from address 412500

Item Names for DASABCIP

The **alarm bits** are bit type items (I/O Discrete for InTouch) and provide VTQ data for VTQ-aware clients: alarm state (0 or 1), timestamp from Allen-Bradley ControlLogix PLC when alarm appears/disappears and item quality. The CLServer item/point naming for alarm bits completely matches with item/point naming used in DASABCIP for Logix5000 controllers (ControlLogix/CompactLogix/FlexLogix) for **Bit within integer** type items, where “integer type item” must be an element of double word (DINT) array (“alarm data array”). In general, the syntax is following:

<Tag_Name>[<element_X>] .bit_#

where:

<Tag_Name> - the name of alarm data array, for example **DB0** or **AlarmDataArray**;

<element_X> - the address of double word integer in the alarm data array , with a valid range **1...number of alarm double words** (element 0 is occupied by alarm data array configuration word);

bit_# - the bit position inside double word, with a range from 0 to 31, where 0 is the least significant bit and 31 is the most significant bit.

The **alarm bit timestamps** are string type items (I/O Message type for InTouch), which can be used additionally to alarm bits; the string will contain the timestamp from ControlLogix PLC when alarm appears/disappears in the following range: from “1990-1-1-0:00:00.000” to “2089-12-31-23:59:59.999”. The item/point name for alarm bit timestamp item needs the suffix **DT** to be added to alarm bit item name:

<Tag_Name>[<element_X>] .bit_#DT

For alarm bit timestamp items, the VTQ timestamp will be the CLServer computer time when string value was provided for client, not ControlLogix PLC actual timestamp.

Item/point naming examples:

DB0[1].0 – I/O Discrete type item, bit 0 in alarm data array “DB” double word with address 1; this is 1st alarm bit in alarm data array “DB0”

DWArray[1].5 – I/O Discrete type item, bit 5 in alarm data array “DWArray” double word with address 1; this is 6th alarm bit in alarm data array “DWArray”

alarms[2].8 – I/O Discrete type item, I/O Discrete type item, bit 8 in alarm data array “alarms” double word with address 2; this is 41st alarm bit in alarm data array “alarms”

alarms[2].8DT – I/O Message type item, bit 8 in alarm data array “alarms” double word with address 2; this is a timestamp for 41st alarm bit in alarm data array “alarms”

The “Demo example program for Allen-Bradley ControlLogix PLC” supports the following I/O items range for CLServer:

DB0[1].0 ... DB0[10].31 (320 alarm bits in array “DB0”)

Item Names for DASOmronEth

The **alarm bits** are bit type items (I/O Discrete for InTouch) and provide VTQ data for VTQ-aware clients: alarm state (0 or 1), timestamp from Omron PLC when alarm appears/disappears and item quality. The CLServer item/point naming for alarm bits completely matches with item/point naming used in DASOmronEth for **bit inside word** type items, where “word type item” must be a word from “alarms area”, located in Data Memory area. In general, the syntax is following:

Dn.b or Dn.bb

where:

n - specifies the Data Memory area word address of the specific word;

b (bb) - specifies the bit number in word (**b** can be 0...9 and **bb** can be 00...15). Number of digits in bit item name with separator is not relevant (leading zeroes are not necessary)

The **alarm bit timestamps** are string type items (I/O Message type for InTouch), which can be used additionally to alarm bits; the string will contain the timestamp from Omron PLC when alarm appears/disappears in the following range: from “1990-1-1-0:00:00.000” to “2089-12-31-23:59:59.999”. The item/point name for alarm bit timestamp item needs the suffix **DT** to be added to alarm bit item name:

Dn.bDT

For alarm bit timestamp items, the VTQ timestamp will be the CLServer computer time when string value was provided for client, not Omron PLC actual timestamp.

Item/point naming examples:

(assuming Omron PLC “alarms area” starts from D1000 and contains 2 alarm words (32 alarm bits) D1005 and D1006)

D1005.0 – I/O Discrete type item, bit 0 in D1005; this is 1st alarm bit in “alarms area”

D1005.5 – I/O Discrete type item, bit 5 in D1005; this is 6th alarm bit in “alarms area”

D1006.4 – I/O Discrete type item, bit 4 in D1006; this is 21st alarm bit in “alarms area”

D1006.4DT – I/O Message type item, bit 4 in D1006; this is a timestamp for 21st alarm bit in “alarms area”

The “Demo example program for Omron CS1 PLC” supports the following I/O items range for CLServer:

D1005.0 ... D1005.15 (16 alarm bits)

Monitoring and Controlling Communication with DAServers

For each topic, there are the following additional items offered by CLServer to monitor and control the communication with corresponding DAServer.

STATUS

For each topic, there is a built-in discrete item that indicates the state of communication between corresponding DAServer and PLC (as well it indicates CLServer and DAServer

communication state). The discrete item (**STATUS**) is set to **0** when communication fails and set to **1** when communication is successful.

From **InTouch** the state of communication may be read by defining an I/O Discrete tagname and associating it with the topic configured for the DAServer corresponding object and using **STATUS** as the item name.

From **Excel**, the status of the communication may be read by entering the following formula in a cell:

=CLSERVER|topic!STATUS

ITEMCOUNT

The **ITEMCOUNT** item is an Integer type Read Only item used to access the number of active items in the corresponding topic, including the **ITEMCOUNT** itself. The range of valid values is from 1 to 2147483647.

ERRORCOUNT

The **ERRORCOUNT** item is an Integer type Read/Write Only (unique for each topic) used to indicate/reset the number of OPC items with errors (received bad quality values occurred since topic initialization or last reset). The **ERRORCOUNT** value can be reset by writing 0 to this item. The range of valid values is from 0 to 2147483647.

ERRORITEMS

The **ERRORITEMS** item is an Integer type Read Only (unique for each topic) used to indicate the number of invalid items which refers to invalid Data Block (for DASSIDIRECT), data array (DASABCIP) or wrong DM area address (DASOmronEth).

WATCHDOG

The **WATCHDOG** item is an Integer type Read Only (unique for each topic) used to indicate the number of seconds since topic activation. The value range is from 0 to 59.

READTIME

The **READTIME** item is an Integer type Read Only (unique for each topic) used to indicate the number of milliseconds spent to read alarm data.

WRITETIME

The **WRITETIME** item is an Integer type Read Only (unique for each topic) used to indicate the milliseconds between writing MSB and writing status confirmation.

Using the CLServer with Suite Link and DDE Clients

The CLServer is accessible from Suite Link clients (e.g. Wonderware InTouch) and DDE clients (e.g. MS Excel).

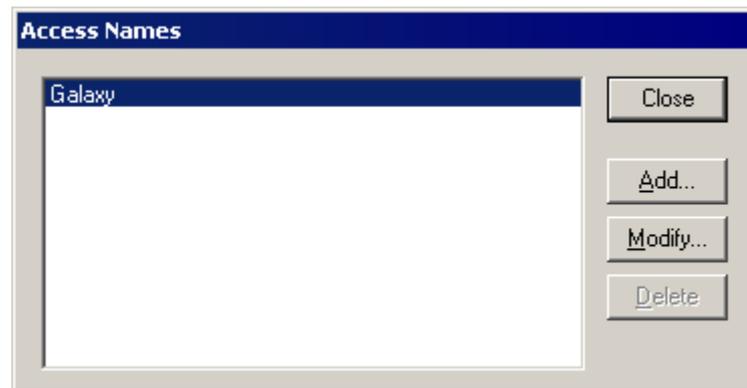
Using the CLServer with InTouch

To access CLServer alarm bits from **InTouch**, the Access Names and Tag names should be defined in **WindowMaker**.

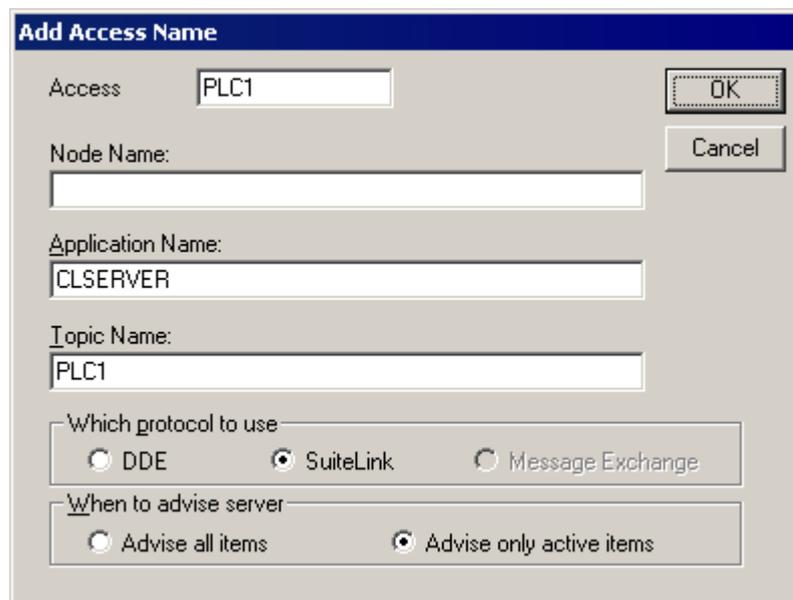
Defining the Access Names

InTouch uses **Access Names** to reference real-time I/O data. Each Access Name equates to an I/O address, which can contain a **Node**, **Application**, and **Topic**. In a distributed application, I/O references can be set up as global addresses to a network I/O Server or local addresses to a local I/O Server.

To define the Access Names in WindowMaker node invoke the `/Special/Access Names...` command. The "Access Names" dialog box will appear.



Click on **A**dd.... The "Add Access Name" Dialog Box will appear:



Note: If **A**dd is selected, this dialog box will be blank when it initially appears. Data has been entered here to illustrate the entries that are made.

The following fields are required entries when entering an Access Name Definition:

Access Name

In the Access Name box type the name you want InTouch to use to this Access Name. (For simplicity, use the same name that you will use for the **Topic Name** here.)

Node Name

If the data resides in a network I/O Server, in the Node Name box, type the remote node's name.

Application Name

In the Application Name box, type the actual program name for the I/O Server program from which the data values will be acquired. In case the values (alarm bits) are coming from S7, ControlLogix or Omron PLCs, the CLSERVER is used. Do not enter the .exe extension portion of the program name.

Topic Name

Enter the name defined for the topic in the CLServer to identify the topic the Server will be accessing.

The Topic Name is an application-specific sub-group of data elements. In the case of data coming from CLServer program, the topic name is the exact same name configured for the topic in the Server.

Note: This will usually be the same as the "Access Name", although, if desired, they may be different. However, it must be the same name used when the topics were configured in **Configuring the CLSERVER** section.

Which protocol to use

Select the **Suite Link** protocol. The DDE is unusable for CLServer as there is no VTQ (Value Time Quality) support in DDE protocol.

When to advise server

Select **Advise all items** if you want the Server program to poll for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. Selecting this option will impact performance, therefore its use is not recommended.

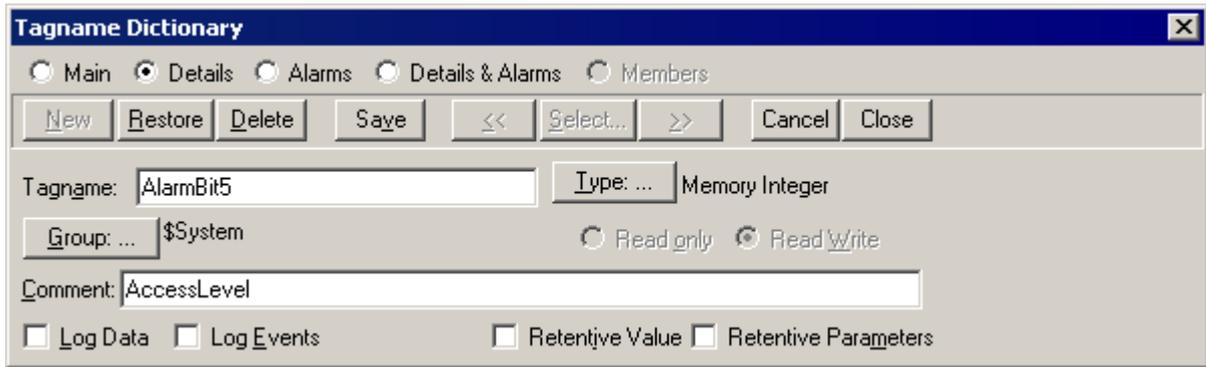
Select **Advise only active items** if you want the Server program to poll only points in visible windows and points that are alarmed, logged, trended or used in any script.

Click **OK** to accept the new Access Name and close the "Add Access Name" dialog box. The "Access Names" dialog box will reappear displaying the new Access Name selected in the list.

Click **Close** to close the "Access Names" dialog box.

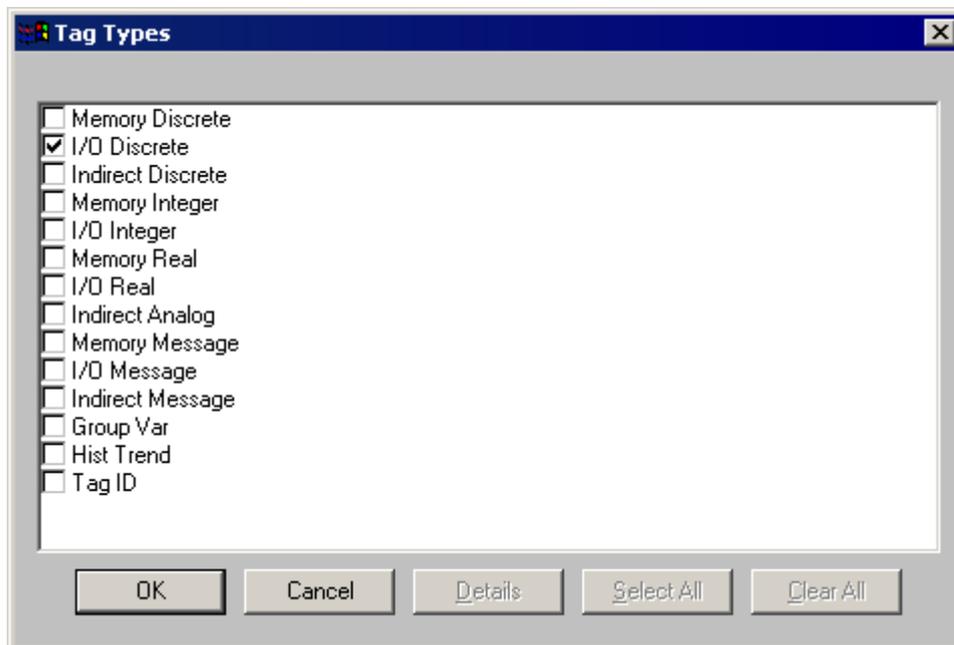
Defining the Tag names

To define the Tag names associated with the new "Access Name", invoke the */Special/Tagname Dictionary...* command (in **WindowMaker**). The "Tagname Dictionary" dialog box will appear:



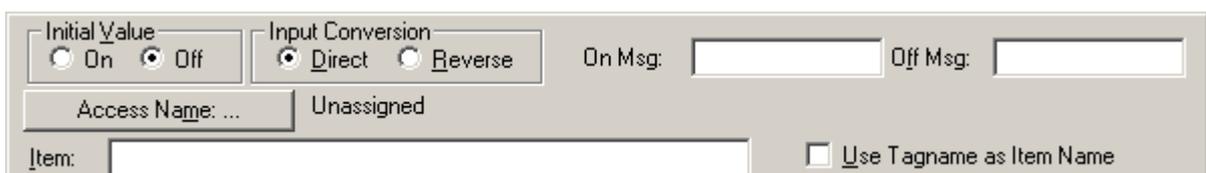
Click on **New** and enter the **Tag Name**. (The tag name defined here is the name **InTouch** will use. The CLServer does not see this name.)

Select the tag type by clicking on the **Type:...** button. The "Tag Types" dialog box will appear:

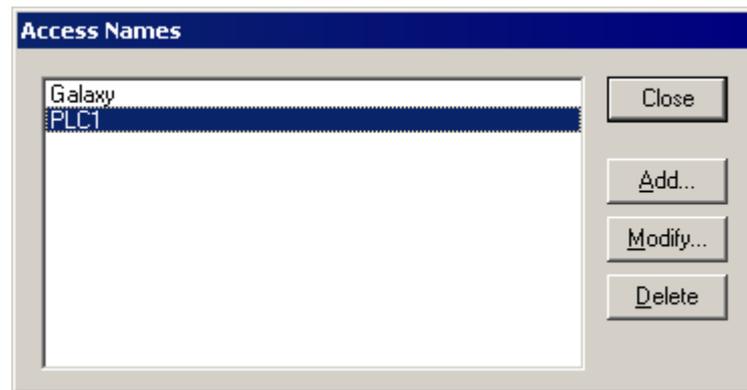


To access alarm bits from PLC via the CLServer, the type must be **I/O Discrete**. Select the Tag type.

The "Details" dialog box for the tag name will appear:



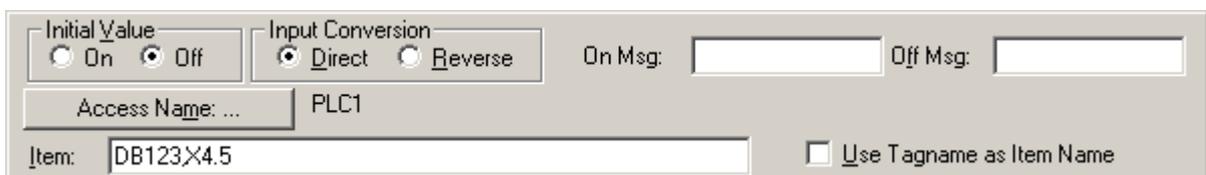
Select the Access Name for CLServer by clicking on the **Access Name:...** button. The "Access Names" dialog box will appear:



Select the appropriate Access Name and click on **Close**. (If the Access Name has not been defined as previously described, click on **Add** and define the Access Name now.) The "Details" dialog box will appear displaying the selected Access Name:



Enter a CLServer item name to be associated with this tagname in the **Item:** field in the "Details" box:



(Refer to the **Item Names** section for complete details.)

Once all entries have been made, click on the **Save** button (in the top dialog box) to accept the new tagname. To define additional tagnames click on the **New** button. To return to the **WindowMaker** main screen, select **Close**.

Monitoring the Status of Communication with InTouch

InTouch supports built-in topic names called **DDEStatus** and **IOStatus**, which are used to monitor the status of communication between the Server and InTouch. For more information on the built-in topic names DDEStatus and IOStatus, see your online "InTouch User's Guide".

The status of communication between the Server and InTouch can be read into **Excel** by entering the following DDE reference formula in a cell on a spreadsheet (in following examples **PLC1** is the Topic Name configured for CLServer):

=view|DDEStatus!PLC1 or **=view|IOStatus!PLC1**

Notes on Using Microsoft Excel

Data from CLServer topics (PLCs) may be accessed from Excel spreadsheets. To do so, enter a formula like the following into a cell on the spreadsheet.

=CLSERVER|topic|item

Sometimes, Excel requires the topic and/or item to be surrounded by apostrophes.

In the formula, **topic** must be replaced with one of the valid topic names defined during the Server configuration process. Replace **item** with one of the valid item names described in the **Item Names** section.

Reading Values into Excel Spreadsheets

Values may be read directly into Excel spreadsheets by entering a DDE formatted formula into a cell, as shown in the following examples:

```
=CLSERVER|'S7PLC'|'DB123,X4.5'  
=CLSERVER|'Logix555'|'alarms[2].8'  
=CLSERVER|'CS1'|'D1005.5'
```

Note: Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

Troubleshooting

WIN.INI entries

The first time you run the CLServer configuration, most of the items in the following list will automatically appear in the WIN.INI file, located in the MS Windows system directory (e.g. C:\Windows). It is an ASCII file and can be altered manually if you wish with any text editor, e.g., MS Windows Notepad (*do not use a program that formats text, such as MS Word or Write unless the file is saved as a DOS text*). The following is a typical entry for the CLServer:

```
[CLSERVER]  
WinIconic=0  
WinFullScreen=0  
WinTop=112  
WinLeft=0  
WinWidth=200  
WinHeight=168  
ShowEvents=0  
ShowOPCCalls=0  
ShowOPCData=0  
ShowData=0  
ShowErrors=1  
Verbose=0  
DebugMenu=1
```

Troubleshooting menu

The following debugging choices are appended to the Server's System Menu (the menu that appears when you click on the Server icon in the upper left hand corner of the Server window):

- ShowEvents** - if checked then Server specific events are logged (not yet implemented).
- Show OPC Calls** - if checked then activity of OPC interface function calls is displayed.
- Show OPC Data** - if checked then OPC data received (OPC callbacks) is displayed.
- Show Alarm Data** - if checked then at any alarm state change all current alarm information from PLC(s) is logged (use with care and for short time as produces very great amount of information logged).
- Dump** - displays all information about Server current internal data structures (topics, items, etc.). (not yet implemented)
- DumpScreen** - if checked then information about active active OPC Server(s), topics and corresponding alarm DBs is displayed on CLServer main window.
- ClearScreen** - if checked then CLServer main window is cleaned.

All debugging information (except **DumpScreen** and **ClearScreen** is displayed via the Wonderware SMC (System Management Console) Log Viewer, which must be active for these commands to work.

Warning: if you check **Show OPC Calls**, **Show OPC Data** and/or **Show Alarm Data**, debug output grows very fast.

KLINKMANN AUTOMATION
CL Server/Client
Revision History

Aug 2005	Rev 1.0	First Release
Jun 2006	Rev 1.1	Support for Allen-Bradley ControlLogix (DASABCIP) and Omron (DASOmronEth) PLCs added.
Jun 2012	Rev 1.2	Modicon/DASMBTCP support added.
Sep 2012	Rev 1.3	“Installing the CLSERVER” section modified, “Licensing by using HASP HL key” and “Software license key” sub-sections added.