Technical Information

OPC interface package
DAQOPC for DX/MV series
DAQOPC for Darwin series
CONTENTS

1. What is OPC ? ........................................................................................................... 1-1

2. What is DAQOPC for DX/MV series ? ........................................................................ 2-1
   2.1 Functions Overview of DAQOPC for DX/MV series .............................................. 2-3
   2.2 System Configuration of DAQOPC for DX/MV Series ........................................... 2-10

3. What is the DAQOPC for DARWIN Series? .......................................................... 3-1
   3.1 Functional Overview of DAQOPC for DARWIN Series ......................................... 3-3
   3.2 System Configuration of DAQOPC for DARWIN Series ....................................... 3-11

4. OPC Client Tool (Supplied with DAQOPC) ......................................................... 4-1
   4.1 Accessing from DAQOPC Client Tool ................................................................. 4-2
   4.2 Utilizing the DAQOPC Client Tool Efficiently ..................................................... 4-10
   4.3 Exchanging the Data between OPC Client and DX/MV/DARWIN ...................... 4-10
   4.4 Trouble Location ................................................................................................. 4-10
   4.5 Trouble between DAQOPC and DX/MV/DARWIN ............................................ 4-10
   4.6 Trouble in the OPC Interface .............................................................................. 4-12

5. Creating an Application Program Using Excel .................................................. 5-1
1. What is OPC?

♦ OPC = OLE for Process Control
  - Standard process data interface for PA and FA industries
    Interface specification for connection between data supplier and data user
  - OLE for process control
    OLE: Object Linking & Embedding
    Object-oriented system provides standard framework for the configuration of reusable software components.
    Standard means for application-to-application communications in Windows environment.

♦ OPC Foundation
  - International Non-Profit Foundation for the specification of the OPC interface
1. What is OPC?

Objective of OPC

♦ Objective of OPC: Neo Industrial Revolution

- To build real multi-vendor open system.
  To integrate various applications written in different languages and running on separate platforms.

- To make multi-layered integration of information exchange on Enterprise Resource Planning (ERP) and Manufacturing Execution System (MES, SCADA, Control) horizontal and flat.

How does Yokogawa Relate to OPC Foundation?

♦ OPC Foundation established 1996
♦ Yokogawa joined OPC Foundation
  - Working as a member of committees for OPC spec definition in all categories:
    • Data Access (DA)
    • Alarms & Events (A&E)
    • Historical Data Access (HDA)
    • (Batch)
    • (Security)
    • (Demo)

♦ Yokogawa also joined OPC Foundation Japan
  - Performing a leading role in OPC-Japan’s steering committee, the Technical Committee
1. What is OPC?

OPC Foundation Members

- OPC (OLE for Process Control)
- OPC Foundation
  - (International Non-Profit Foundation)
  - OPC Council
    - Japan
    - About 50 companies
  - OPC Council
    - Europe
    - About 50 companies
  - About 260 companies worldwide

Issues up to now

User Needs → Information System
- Data can be shared among information system computers.
- Corporate top executives can effectively access and use plant info to satisfy business needs in a timely manner.

Issues up to now
- DX/MV, DARWIN data could not be shared with supervisory computer.
- DX/MV, DARWIN measurement data was unusable by supervisory computer if the format remains as it was.
- DX/MV, DARWIN data scattered like “isolated islands of datum”
1. What is OPC?

**Previous Open Interface**

- **Needed vendor-specific interface**

<table>
<thead>
<tr>
<th>Application A</th>
<th>Application B</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
</tr>
</tbody>
</table>

- **Needed each Recorder-specific interface**

- **Needed each PCS-specific interface**

**Diagram:**

- A: DX, MV
- B: DARWIN
- C: Recorder of other make

**Solutions to Issues**

**Introduction of OPC Interface**
- Both DX/MV, DARWIN and application support standard interface
- Operation data and corporate data can be integrated

**Here at Yokogawa**
- We perform a leading role in OPC spec committees.
- OPC Interface has already supported on DAQOPC for DX/MV and DAQOPC for DARWIN.

- Ahead to products of other makes, DAQOPC has satisfied customers’ needs.
Open Interface using OPC

Connectable with single interface

Minimized Development Cost

Application A (OPC Client)

 OPC Interface

Application B (OPC Client)

Application Reusability

DX, MV

DARWIN

Recorder of Other Make

Merits of OPC

♦ For User
- Wide selection
  Flexible system integration
- Cuts the total cost
  System integration

♦ For Application Engineer
- Using general-purpose software tool
- Unnecessary to develop original interface
- Time and cost reduction
  Reusing existing application

♦ For Development Engineer
- Easy connection
- Time and cost reduction
  Reusing existing driver
2. What is DAQOPC for DX/MV series?

The OPC interface* is the standard for process data exchange in the PA and FA industries. The OPC (OLE for Process Control) interface is the OPC standard interface of which the specifications are defined by the OPC Foundation.

* OPC (OLE for Process Control)

- Data Access (DA) server function
- Process data with the extension of item ID can be loaded to/from the server.
- Browsing function
- The contents in the OPC server can be browsed by the OPC clients.

DAQOPC (OPC Server)

LAN

DX100
MV100
MV200
DX200

DX/MV (up to 24 units connectable)
What is the DAQOPC for DX/MV Series?

An interface package conforming to the OPC standard interface designed for accessing to the DX/MV unit from an application (OPC client) compliant with OPC. Conforming to the OPC Standard Interface

- Data Access interface (DA Server)
- Reading/writing the process data measured by DX/MV units
2.1 Functions Overview of DAQOPC for DX/MV series

Data Access (DA) Server Features
- Reads & Writes process data using item ID as data identifier
  - Read process data
  - Synch, Asynch: Refresh/Subscription (Data change notices)
  - Write process data
  - Synch, Asynch

OPC Interface-compliant Functions
- OPC Data Access Custom Interface Specification Version 2.0
- OPC Data Access Automation Specification Version 2.0

Program ID to access data
- Server name: Yokogawa.ExaopcDADAQOPCDXMV1
Basic Configuration of DAQOPC for DX/MV Series

Setting the parameters of DAQOPC

- Define a group object from the OPC client to the server object.
- Register the process data to be accessed as an item ID in the group object.
- Register the item IDs in the form of "device name!tag name.data item name".

Data Access Method (1)

- [Sever Object]
  - Sever object A

- [Group Object]
  - Trend 1
  - Report 1

- [Item ID]
  - Device1.ch01.PV
  - Device1.ch02.PV
  - Device1.ch03.PV
  - Device1.ch04.PV
  - Device1.ch01.Alarm1
# Item ID / Data Type

<table>
<thead>
<tr>
<th>Item ID</th>
<th>Description</th>
<th>Variant data type</th>
<th>Access type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVICE*! CH###.PV</td>
<td>Measured data, results of computation</td>
<td>VT_R8</td>
<td>Read only</td>
<td>Measurement/computation channels only</td>
</tr>
<tr>
<td>DEVICE*! CH###.ALARMx</td>
<td>Alarm status of the specified channel</td>
<td>VT_I2</td>
<td>Read only</td>
<td>Measurement/computation channels only</td>
</tr>
<tr>
<td>DEVICE*! CH###.INPUT</td>
<td>Communication input data value</td>
<td>VT_R8</td>
<td>Read/Write</td>
<td>Communication input data only</td>
</tr>
<tr>
<td>DEVICE*! COMMON.STATUS</td>
<td>Status of the device</td>
<td>VT_I2</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>DEVICE*! COMMONPROP</td>
<td>Attribute read status</td>
<td>VT_I2</td>
<td>Read/Write</td>
<td></td>
</tr>
</tbody>
</table>

*: Device No.  
###: Channel No.  
Type of variant data:  
VT_R8: 8 bytes (double-precision floating-point number)  
VT_I2: 2 bytes (single-precision integer)

## Data Item Names and Specifiable Tag Names

- **PV**  
  Specifiable tag name: Existing measurement/computation channels only
- **ALARM**  
  Specifiable tag name: Existing measurement/computation channels only  
  The item name must be ALARM1, ALARM2, ALARM3 or ALARM4.  
  When no alarm has occurred: 0  
  When an alarm has occurred: The alarm type which is the item attribute of the alarm ("alarm type" is a value ranging from 1 to 8 which represents the alarm types listed in the description of property ID: 6004.) (See page 2-14 in the IM.)
- **INPUT**  
  Specifiable tag name: Existing communication input data only  
  The range of the value which can be entered is that of the DX/MV.  
  The server does not check whether the entered value is within the available range or not.
- **STATUS**  
  Specifiable tag name: COMMON  
  The status of the specified device is as follows:  
  Normally operating: 0  
  The specified device does not exist: 1  
  Failed to secure the memory area: 2  
  Failed to establish the communication line: 3  
  Communication error: 4  
  If "0" is written while the status is "3" or "4", the system attempts to restore the communication line of the specified device.  
  If "0" is written while the status is "1" or "2", it has no effect.
Prop
Specifiable tag name: COMMON
The attribute read status of the specified device is as follows:
  - The attribute information of the specified device has not been read in: 0
  - The attribute information of the specified device has been read in: 1
The attribute information of the specified device is read in: 1
If "2" is written, the attribute information of the specified device is obtained. Normally, the attribute information is read in at the start-up of the DAQOPC for DX/MV series. The response to the request to obtain the property value is made based on this value instead of issuing an obtainment request to the device actually. Also for the decimal point position information of the computation value, the value at the start-up of the DAQOPC for DX/MV series is used. Consequently, it is necessary to perform read-in operation again if you changed the range setting or the like of the specified device.
Data and Quality Code

The data from the DX/MV unit is converted to the following quality codes.

<table>
<thead>
<tr>
<th>Data Status</th>
<th>Major Key</th>
<th>Detailed Information</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>A normal value</td>
<td>Good</td>
<td>Non-specific</td>
<td>Not Limited</td>
</tr>
<tr>
<td>A value exceeded to the + side</td>
<td>Good</td>
<td>Non-specific</td>
<td>High Limited</td>
</tr>
<tr>
<td>A value exceeded to the - side</td>
<td>Good</td>
<td>Non-specific</td>
<td>High Limited</td>
</tr>
<tr>
<td>Skip or no computation</td>
<td>Bad</td>
<td>Out of Service</td>
<td>Not Limited</td>
</tr>
<tr>
<td>An error value</td>
<td>Bad</td>
<td>Sensor Failure</td>
<td>Not Limited</td>
</tr>
<tr>
<td>No data</td>
<td>Bad</td>
<td>Out of Service</td>
<td>Not Limited</td>
</tr>
<tr>
<td>In case of a communication failure</td>
<td>Bad</td>
<td>Comm Failure</td>
<td>Not Limited</td>
</tr>
</tbody>
</table>

Time Information (Time Stamp)

Either the time of the DX/MV unit or that of the PC can be obtained. The time source can be changed by changing the setting in the DAQOPC for DX/MV series.

<table>
<thead>
<tr>
<th>Setting in the DAQOPC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of the device (DX/MV)</td>
<td>The time of the target device can be directly obtained.</td>
</tr>
<tr>
<td>Time of the PC</td>
<td>The time of the PC bearing the server at the time of data acquisition is obtained on a basis of UTC.</td>
</tr>
</tbody>
</table>
2. What is DAQOPC for DX/MV series?

Application Interface

- OPC Custom Interface and Automation Interface
  There are two types of interfaces defined in the OPC specifications: the OPC custom interface and the automation interface. The two interfaces have almost the same data access functions. However, the intended client programs are different. Since the OPC server of the DAQOPC for DX/MV series is provided with both interfaces, it supports both client programs.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Custom Interface</th>
<th>Automation Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>For the special-purpose applications such as SCADA, MES, analyzer software</td>
<td>For the easy access from the script language</td>
<td></td>
</tr>
<tr>
<td>Client’s language</td>
<td>VC++</td>
<td>Visual Basic</td>
</tr>
<tr>
<td>Performance</td>
<td>☐</td>
<td>△</td>
</tr>
</tbody>
</table>

The OPC custom interface is an interface using just the basic functions of the OLE/COM and operated at high speed. The OPC automation interface complies with the OLE automation interface which can be accessed from Visual Basic and other languages. Its performance is slightly poor compared with the OPC custom interface because the processing overhead to facilitate the access from the script language is incurred.

- Versions of VB and VC++
  When an application program (OPC client) is created by the user, use the VB and VC++ of the following versions.

<table>
<thead>
<tr>
<th>DAQOPC and the user AP installed on the same PC</th>
<th>DAQOPC and the user AP separately installed on different PCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAQOPC R1.02 VC6.0, VC++6.0</td>
<td>VB5.0/6.0, VC++5.0/6.0</td>
</tr>
</tbody>
</table>

When creating a program with VB, select [Project (P)] - [References (N)] on VB and mark the checkbox for "Yokogawa Exaopc Data Access Automation Server".
2. What is DAQOPC for DX/MV series?

Connectable Devices

<table>
<thead>
<tr>
<th>Device</th>
<th>Connection Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX100,DX200</td>
<td>Ethernet</td>
</tr>
<tr>
<td>DX100L,DX200C</td>
<td>RS-232C</td>
</tr>
<tr>
<td>MV100,MV200</td>
<td>RS-422A</td>
</tr>
</tbody>
</table>

Capacity of the Application

<table>
<thead>
<tr>
<th>Item</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max number of clients</td>
<td>100</td>
</tr>
<tr>
<td>Max number of group objects</td>
<td>1,000 per groups</td>
</tr>
<tr>
<td>Max number of registered item IDs</td>
<td>10,000 per group</td>
</tr>
<tr>
<td>Max number of cache update item IDs</td>
<td>100,000</td>
</tr>
<tr>
<td>Cache update cycle (Update Rate)</td>
<td>1 to 3,600 sec.</td>
</tr>
<tr>
<td>Max number of networked DX/MV</td>
<td>24</td>
</tr>
</tbody>
</table>
2.2 System Configuration of DAQOPC for DX/MV Series

♦ When the DAQOPC for DARWIN Series and the OPC Client Coexist
  - Application size: medium
  - Performance: High-speed, as it uses a local OPC server
2. What is DAQOPC for DX/MV series?

Sample Client-Server Configuration (2)

- When the OPC Client Exists on the Host Computer
  - Application size: Medium to Large
  - Performance: a bit inferior to co-locating type

Sample Configuration of Multiple Clients

- Accessing from Two or More Number of OPC Clients to One OPC Server
Sample Configuration of Multiple Servers (1)

♦ Accessing from One OPC Client to Two or More Number of OPC Servers

![Diagram of sample configuration](image)

Specifications (1)

♦ System Requirements for Hardware

- **Machine type**: IBM PC/AT compatible (supporting Windows NT or Windows 2000)
- **CPU**: Pentium II 300 MHz or higher
- **Main memory**: 128 MB or greater (with Windows NT) 256 MB or greater (with Windows 2000)
- **Disk space**: 130 MB or greater
- **LAN adapter**: Ethernet-compatible network card

This is not necessary when the OPC client and the DAQOPC are installed on the same PC and the data is obtained from the DX/MV unit via RS-232C/RS-422-A.
Specifications (2)

♦ System Requirements for Software

- **OS**: Windows NT Workstation 4.0/Windows NT Server 4.0 or Windows 2000
  The package (Exaopc) and the OS (Windows NT/2000) must be of the same language.

- **Service pack**: Windows NT service pack 4, 5, 6a
  Windows 2000 service pack 1
3. What is the DAQOPC for DARWIN Series?

- **Data Access (DA) Server Function**
  - The process data with the extension of item ID can be read/write.

- **Browse Function**
  - The contents of the OPC server can be browsed by OPC clients.

*The OPC (OLE for Process Control) interface is the OPC standard interface of which the specifications are developed and maintained by the OPC Foundation.*
What is the DAQOPC for DARWIN Series?

♦ What is the DAQOPC for DARWIN Series?
   An interface package conforming to the OPC standard interface designed for accessing to a DARWIN unit from an application (OPC client) compliant with OPC.

♦ Conforming to the OPC Standard Interface
   Data Access interface (DA Server)
   - Reading/writing the process data measured by DARWIN units
3.1 Functional Overview of DAQOPC for DARWIN Series

♦ Data Access (DA) Server Function
  - Reading/writing the process data with the extension of item ID
    Reading the process data
    Synchronous, asynchronous: refresh/subscription (notification of data change)
    Writing the process data
    Synchronous, asynchronous

♦ OPC Protocol
  - Conforming to the OPC Data Access Custom Interface Specification Version 2.0
  - Conforming to the OPC Data Access Automation Specification Version 2.0

♦ Reading/Writing the Process Data with the Extension of Item ID
  - Name of the server: Yokogawa.ExaopcDADAQOPCDARWIN1
Basic Configuration of the DAQOPC for DARWIN Series

Device name = Device * !
* : No

Setting up the serial communication conditions

Model
Meas C: Measurement channel count
MathCh: Computation channel count
System: System/module configuration
Port: Port type
Ethernet
COM (Serial)

Data Access Method (1)

- Define a group object from the OPC client to the server object.
- Register the process data to be accessed as an item ID in the group object.
- Register the item IDs in the form of "device name!tag name.data item name".

[Sever Object] [Group Object] [Item ID]
Sever object A Trend 1
Device1!ch01.PV
Device1!ch02.PV
Device1!ch03.PV
Device1!ch04.PV
Device1!ch01.Alarm1

Device name!tag name
Data item name
Data Access Method (2)

♦ Item ID / Data Type

<table>
<thead>
<tr>
<th>Item ID</th>
<th>Description</th>
<th>Variant data type</th>
<th>Access type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVICE*!CH###.PV</td>
<td>Measured data, results of computation</td>
<td>VT_R8</td>
<td>Read only</td>
<td>Measurement/computation channels only</td>
</tr>
<tr>
<td>DEVICE*!CH###.ALARMx</td>
<td>Alarm status of the specified channel</td>
<td>VT_I2</td>
<td>Read only</td>
<td>Measurement/computation channels only</td>
</tr>
<tr>
<td>DEVICE*!CH###.INPUT</td>
<td>Communication input data value</td>
<td>VT_R8</td>
<td>Read/Write</td>
<td>Communication input data only</td>
</tr>
<tr>
<td>DEVICE*!COMMON.STATUS</td>
<td>Status of the device</td>
<td>VT_I2</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>DEVICE*!COMMON_PROP</td>
<td>Attribute read status</td>
<td>VT_I2</td>
<td>Read/Write</td>
<td></td>
</tr>
<tr>
<td>DEVICE*!R$$$.STATUS</td>
<td>Command DO status</td>
<td>VT_BOOL</td>
<td>Read/Write</td>
<td>Command DO only</td>
</tr>
</tbody>
</table>

*: Device No.  
###: Channel No.  
$: Relay No.  
x: Alarm level No.  

Type of variant data:  
VT_R8: 8 bytes (double-precision floating-point number)  
VT_I2: 2 bytes (single-precision integer)  
VT_BOOL: Unsigned number, 0: False 1: True

♦ Data Item Names and Specifiable Tag Names

- **PV**
  Specifiable tag name: Existing measurement/computation channels only

- **ALARM**
  Specifiable tag name: Existing measurement/computation channels only
  The item name must be ALARM1, ALARM2, ALARM3 or ALARM4.
  When no alarm has occurred: 0
  When an alarm has occurred: The alarm type which is the item attribute of the alarm ("alarm type" is a value ranging from 1 to 6 which represents the alarm types listed in the description of property ID: 6004.) (See page 3-14 in the IM.)

- **INPUT**
  Specifiable tag name: Existing communication input data only
  The range of the value which can be entered is that of the DARWIN. The server does not check whether the entered value is within the available range or not. Not used when 34151 is used as the port No. for Ethernet.

- **STATUS(Rxxx)**
  Specifiable tag name: Rxxx
  Only the command DO which can be used on the DARWIN DA model (style 8 or later) is specifiable.
  The status of the specified relay is as follows:
  - Relay off: FALSE
  - Relay on: TRUE
  Not used when 34151 is used as the port No. for Ethernet.
### STATUS

* Specifiable tag name: COMMON

The status of the specified device is as follows:
- Normally operating: 0
- The specified device does not exist: 1
- Failed to secure the memory area: 2
- Failed to establish the communication line: 3
- Communication error: 4

If "0" is written while the status is "3" or "4", the system attempts to restore the communication line of the specified device. If the DAQOPC has been set so that a reception is made at start-up, a reception is made at the communication line restoration with the fixed file written and the attribute information obtained.

If "0" is written while the status is "1" or "2", it has no effect.

### PROP

* Specifiable tag name: COMMON

The attribute read status of the specified device is as follows:
- The attribute information of the specified device has not been read in: 0
- The attribute information of the specified device has been read in: 1
- The attribute information of the specified device is read in: 1

If "2" is written, the attribute information of the specified device is obtained. Normally, the attribute information is read in at the start-up of the DAQOPC for DARWIN series.

The response to the request to obtain the property value is made based on this value instead of issuing an obtainment request to the device actually. Also for the decimal point position information of the computation value, the value at the start-up of the DAQOPC for DARWIN series is used. Consequently, it is necessary to perform read-in operation again if you changed the range setting or the like of the specified device.
3. What is the DAQOPC for DARWIN Series?

Data Access Method (3)

♣ Sample Connection (Reading Process Data and Alarm Status)

Client

- Trend
  - Item ID 1
  - Item ID 2
  - Item ID 3
  - Item ID 4

- Report
  - Item ID 1
  - Item ID 5

DAQOPC for DARWIN series

- Device 1
  - Item ID 1
  - Item ID 2
  - Item ID 3
  - Item ID 4
  - Item ID 5

- Device 1 DARWIN

YOKOGAWA ©

OPC Specification Process Data

- Data Value: Value
- Quality Code: Quality Flag
- Time Information: Time Stamp

Bit

From Limited/LL/HL

Good/Bad/Uncertain

Not Limited/LL/HL

Network Solutions PMK

Detailed information on a quality classification code basis

For the code table, see page 3-8 in the IM.
Data and Quality Code

The data from the DARWIN unit is converted to the following quality codes.

<table>
<thead>
<tr>
<th>Data Status</th>
<th>Major Key</th>
<th>Detailed Information</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>A normal value</td>
<td>Good</td>
<td>Non-specific</td>
<td>Not Limited</td>
</tr>
<tr>
<td>A value exceeded to the + side</td>
<td>Good</td>
<td>Non-specific</td>
<td>High Limited</td>
</tr>
<tr>
<td>A value exceeded to the - side</td>
<td>Good</td>
<td>Non-specific</td>
<td>High Limited</td>
</tr>
<tr>
<td>Skip or no computation</td>
<td>Bad</td>
<td>Out of Service</td>
<td>Not Limited</td>
</tr>
<tr>
<td>An error value</td>
<td>Bad</td>
<td>Sensor Failure</td>
<td>Not Limited</td>
</tr>
<tr>
<td>No data</td>
<td>Bad</td>
<td>Out of Service</td>
<td>Not Limited</td>
</tr>
<tr>
<td>In case of a communication failure</td>
<td>Bad</td>
<td>Comm Failure</td>
<td>Not Limited</td>
</tr>
<tr>
<td>When the specified item does not exist</td>
<td>Bad</td>
<td>Configuration Error</td>
<td>Not Limited</td>
</tr>
</tbody>
</table>

Time Information (Time Stamp)

Either the time of the DARWIN unit or that of the PC can be obtained. The time source can be changed by changing the setting in the DAQOPC for DARWIN series.

<table>
<thead>
<tr>
<th>Setting in the DAQOPC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of the device (DARWIN)</td>
<td>The time of the target device can be directly obtained.</td>
</tr>
<tr>
<td>Time of the PC</td>
<td>The time of the PC bearing the server at the time of data acquisition is obtained on a basis of UTC.</td>
</tr>
</tbody>
</table>

If the target device is a DA100 unit, the time of the PC is used even if "Time of the device (DARWIN)" has been specified.

Application Interface

- OPC Custom Interface and Automation Interface

There are two types of interfaces defined in the OPC specifications: the OPC custom interface and the automation interface. The two interfaces have almost the same data access functions. However, the intended client programs are different. Since the OPC server of the DAQOPC for DARWIN series is provided with both interfaces, it supports both client programs.

<table>
<thead>
<tr>
<th>Custom Interface</th>
<th>Automation Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>For the special-purpose applications such as SCADA, MES, analyzer software</td>
</tr>
<tr>
<td>Client’s language</td>
<td>Visual Basic</td>
</tr>
<tr>
<td>Performance</td>
<td>⬤</td>
</tr>
</tbody>
</table>

The OPC custom interface is an interface using just the basic functions of the OLE/COM and operated at high speed. The OPC automation interface complies with the OLE automation interface which can be accessed from Visual Basic and other languages. Its performance is slightly poor compared with the OPC custom interface because the processing overhead to facilitate the access from the script language is incurred.
3. What is the DAQOPC for DARWIN Series?

- **Versions of VB and VC++**
  When an application program (OPC client) is created by the user, use the VB and VC++ of the following versions.

<table>
<thead>
<tr>
<th>DAQ OPC</th>
<th>DAQOPC and the user AP installed on the same PC</th>
<th>DAQOPC and the user AP separately installed on different PCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1.02</td>
<td>VB 6.0, VC++ 6.0</td>
<td>VB 5.0/6.0, VC++ 5.0/6.0</td>
</tr>
</tbody>
</table>

When creating a program with VB, select [Project (P)] - [References (N)] on VB and mark the checkbox for "Yokogawa DAQOPC Data Access Automation Server".

**Connectable Devices**

<table>
<thead>
<tr>
<th>Device</th>
<th>Connection Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA100, DC100, DR130</td>
<td>Ethernet</td>
</tr>
<tr>
<td>DR23, DR232, DR241</td>
<td>RS-232C</td>
</tr>
<tr>
<td>DR242</td>
<td>RS-422A</td>
</tr>
</tbody>
</table>
### Capacity of the Application

<table>
<thead>
<tr>
<th>Item</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of clients</td>
<td>100 clients</td>
</tr>
<tr>
<td>Maximum number of group objects</td>
<td>1,000 groups</td>
</tr>
<tr>
<td>Maximum number of registrable item IDs</td>
<td>10,000/group</td>
</tr>
<tr>
<td>Maximum number of cache refresh item IDs</td>
<td>100,000</td>
</tr>
<tr>
<td>Cache refresh rate</td>
<td>1 to 3,600 seconds</td>
</tr>
<tr>
<td>Maximum number of connectable DX/MV units</td>
<td>16</td>
</tr>
</tbody>
</table>
3.2 System Configuration of DAQOPC for DARWIN Series

♦ When the DAQOPC for DARWIN Series and the OPC Client Coexist
  - Application size: Medium
  - Performance: At high speed due to the OPC server installed locally
Sample Client-Server Configuration (2)

♦ When the OPC Client Exists on the Host Computer
  - Application size: Medium to large
  - Performance: Slightly poor compared to the coexistent type

Sample Configuration of Multiple Clients

♦ Accessing from Two or More Number of OPC Clients to One OPC Server
Sample Configuration of Multiple Servers (1)

❖ Accessing from One OPC Client to Two or More Number of OPC Servers

![Diagram showing network configuration]

Specifications (1)

❖ System Requirements for Hardware

- Machine type: IBM PC/AT compatible (supporting Windows NT or Windows 2000)
- CPU: Pentium II 300 MHz or higher
- Main memory: 128 MB or greater (with Windows NT) 256 MB or greater (with Windows 2000)
- Disk space: 130 MB or greater
- LAN adapter: Ethernet-compatible network card

This is not necessary when the OPC client and the DAQOPC are installed on the same PC and the data is obtained from the DX/MV unit via RS-232C/RS-422-A.
Specifications (2)

♦ System Requirements for Software

- **OS**: Windows NT Workstation 4.0/Windows NT Server 4.0 or Windows 2000
  The package (Exaopc) and the OS (Windows NT/2000) must be of the same language.

- **Service pack**: Windows NT service pack 4, 5, 6a
  Windows 2000 service pack 1
4. OPC Client Tool (Supplied with DAQOPC)

The following describes how to operate the following software products of OPC Client Tool, which are installed at the same time when DAQOPC is installed.

- ZOPDAClient.exe
- ZOPDAVBTTest.exe

Note that Yokogawa Electric Corp. does not accept any complaints about OPC Client Tool described in the following since this software product is beyond the support of DAQOPC.
4.1 Accessing from DAQOPC Client Tool

DAQOPC Client Tool provides two kinds of software products, ZOPDAClient.exe and ZOPDAVBT Test.exe. Since both software products have similar operating procedures, the following mainly describes ZOPDAClient.exe.

ZOPDAClient.exe exists in the following folder. “Install folder”¥exaopc¥tool”
The default “Install folder” is “C:¥DAQOPC” and the following screen shows the startup screen.

The following describes the operating procedures.

(1) Connecting to DAQOPC
To connect to DAQOPC, select the ProgID of DAQOPC connected using the ProgID in the “Connect” frame at the upper left portion of the screen.

DAQOPC for DX/MV Series is Yokogawa.ExaopcDADAQOPCDXMV1. DAQOPC for DARWIN is Yokogawa.ExaopcDADAQOPCDARWIN1.

When connecting to DAQOPC in the remote mode, set the Node name or IP Address of the PC where DAQOPC exists for “Node” above “ProgID”. They do not need to set when connecting to DAQOPC in the local mode.

CAUTION! To connect to DAQOPC in the remote mode, it is absolutely necessary to run the DAQOPC Client setup on the PC where the Client software exists. For details, see IM.

Click the “Connect” button in the “Connect” frame to connect to DAQOPC you have selected. When connected successfully, various kinds of information are shown in the “Server Information” frame located under the “Connect” frame. The details about display contents are described later.

(2) Creating a group in DAQOPC
Input a group name in “Group” in the “Select Group” frame on the right of the “Connect” frame. You may input a desired group name. If you do not input any group name, the default group name is assigned.

Click the “Select/Add” button.
The information about the group is shown in the “Group Information” frame under the “Select Group” frame.
(3) Registering an item ID to the group

Input the item ID of the data you wish to obtain in the “Item Management” frame at the lower left potion of the “Group Information” frame. (The item IDs registered in the group are shown in the list. To register an item ID, edit the text box next to “Item”.)

Click the “Add” button in the “Item Management” frame.

The item ID you have input is shown in the “Item ID” column of the spreadsheet located under the “Item Management” frame. If you have input an incorrect item ID, relevant error will appear.

It is also possible to add the item ID of the data you wish to obtain in the same manner as described above.

(4) Obtaining data

Click the “Sync Read” button in the “Read/Write” frame next to the “Item Management” frame.

The obtained data is shown in the “Value” columns next to “Item ID” in the spreadsheet.

The above describes the simple data obtaining procedures.

Next, each item of ZOPDAClient.exe is described in detail.

[Connect frame] Connects or disconnects the OPC Server.

[Node] Specify the name or IP address of the PC where the OPC Server exists if the OPC Server you wish to connect does not exist in the same PC. If the OPC Server exists in the same PC, it is not necessary to specify the name or IP address.

To connect to DAQOPC in other PC, it is absolutely necessary to run the Client setup on the PC where ZOPDAClient.exe is running. For details, see IM. However, it is not necessary to run the Client setup if DAQOPC is installed on the PC where ZOPDAClient.exe is running.

[ProgID] Select the ProgID of a OPC Server you wish to connect. If DAQOPC you wish to connect is not shown in the selection list, the Client setup has not been run. In this case, run the Client setup.

[Connect] Connects to the OPC Server specified by [Node] and [ProgID]. When connected successfully, the information about Server is shown in the “Server Information” frame.

[Disconnect] Disconnects the connection to the OPC Server connected currently.

[Server Information frame] Shows the information about the connected OPC Server.

[StartTime] Shows the time when the OPC Server is started up.

[CurrentTime] Shows the current time of the OPC Server. Clicking the “GetStates” button will update this time.

[LastUpdateTime] Shows the latest data update time of the OPC Server. Clicking the “GetStates” button will update this time.
[MajorVersion] Shows the major version of the OPC Server. “1” is shown for DAQOPC.

[MinorVersion] Shows the minor version of the OPC Server. “0” is shown for DAQOPC.

[BuildNumber] Shows the build number of the OPC Server. “1” is shown for DAQOPC.

[VendorInfo] Shows the information about the vendor of the OPC Server. “Yokogawa Electric Corporation” is shown for DAQOPC.

[ServerState] Shows the status of the OPC Server. “1” is shown for DAQOPC.

[BandWidth] Shows the bandwidth of the OPC Server. The number of registered item IDs is shown for DAQOPC.

[ClientName] Shows no data for DAQOPC.

[SetLocale] You may specify the LocaleID of the OPC Server. This item is invalid for DAQOPC.

[SetUser] Shows no data for DAQOPC.

[Level] Shows “0” for DAQOPC.


[Browse] Shows the item IDs available for the OPC Server. For details, see the next section.

[Property] Shows the attribute values of the item IDs available for the OPC Server. For details, see the next section.

---

[Browse dialog box] Shows the item IDs available for the OPC Server.

The screen display uses the tree structure. When clicking a desired item, you may check available item IDs.
[Organization] Shows the Browse type supported by the OPC Server. “1” is shown for DAQOPC.

[Filter] Limits the display contents by filtering. This function is not supported by DAQOPC.

[Current Position] Shows the item name at the focus position of the tree.

[Delimiter] Shows the delimiter at the focus position of the tree.

In DAQOPC, “!” is shown when the device name is displayed in “Current Position” and “.” is shown when the channel name is displayed in “Current Position”.

[Get Property] Shows the “Properties” dialog box to display the attributes of the contents shown in [Current Position].

[Close] Closes the “Browse” dialog box.

---

**Properties dialog box** Obtains the information about the attribute of the specified item name. The OPC Server returns the property values of the specified item name in respect to the request of the client.

Using the specified item name property ID, the client can request the server to obtain the property values.

In DAQOPC, the property ID, which can be specified, may vary depending on the item. For details, see IM.
[AvailablePropertyIDs]
Input an item ID in “ItemID” and click the “QueryProperties” button. The list of property IDs, which can be obtained by the specified item ID, are displayed. For details about meanings of property IDs, see IM.

[Properties]
Input an item ID in “ItemID” and input a property ID you wish to obtain for this item ID in “ID”, and then click the “QueryProperties” button. The data of the specified contents is displayed. If you leave “ID” blank, the information about all attributes of the specified item IDs can be obtained.

[Select Group frame] Adds or deletes a group to/from the connected OPC Server, or changes the contents of the currently displayed group.

[Group]
Input a new group name if you wish to add a group. Change the group name if you wish to delete the group or change the display group.

[Select/Add]
Changes the displayed group to the group specified in “Group”. If the specified group exists, its contents are shown in the frame located under “Select/Add”. If not, the contents of a new group are displayed.

[Delete]
Deletes the group specified in “Group”.

[Group Information frame] Shows the contents of the group specified in the “Select Group” frame.

[Name] Shows the name of the specified group.

[ServerHandle] Shows the group identifier put on the server for the specified group.

[ClientHandle] Shows the group identifier put on the client for the specified group.

[LCID] Shows the LocaleID used for the specified group.

[UpdateRate] Shows the data update interval (in msec.) of the specified group.

[TimeBias] Shows the time bias of the specified group.

[DeadBand] Shows the width of the dead band when running “Subscribe” for the data of the specified group. A value in % to the span value of each item ID is shown.

[Active] Specifies whether or not the specified group is made valid.

[Subscribe] Specifies whether or not the subscribe is run for the specified group. Subscribe means the data change notification. When running the subscribe for AsyncRead, the server sends the data change notification to the client if a variation exceeding that specified in “DeadBand” occurs in the registered item ID. This function is limited to the real data type. For values other than real values, the change notification is performed regardless of DeadBand if the value varies. This function is intended to avoid the noise of the item ID.

[Enable] Specifies whether or not the subscribe is valid for the specified group.

[SetStates] Sets the information about the group information to the server. This is intended to set the values specified in “UpdateRate” and “DeadBand”.

[GetStates] Obtains the information about the group information.
[Item Management frame] Adds or deletes an item ID to/from the specified group.
   Input a new item ID if you wish to add it. Select an item ID if you wish
to delete it.

[Type] Specify a data type of the item ID you wish to add, which is requested from the
client. Each item ID has a data type specified by the server. For details, see IM.
This data type is called “Canonical data type”. When the data type (called
“request data type”) is specified from the client, the server converts the
original data type into that specified by the client, and transfers it to the client.
However, note that some data types cannot be converted.
If “VT_EMPTY” is specified for the request data type, the canonical data type
value is returned.

[Add] Adds the item ID to the group according to the contents specified in “Item” and
“Type”. The added item ID will affect the spreadsheet.

[Delete] Deletes the selected item ID from the group. The deleted item ID will be deleted
from the spreadsheet.

[Read/Write frame] Specify data operation.

[Data Source] Select “Device” or “Cache”. If you specify “Device”, both the reading and
writing directly access the recorder through the server.
If you specify “Cache”, the server returns the cache data in the server, which is
updated at specified intervals. Therefore, it is thought that use of “Cache”
becomes the high-speed operation when viewed from the client.

[Sync Read] Starts the synchronous read process.
[Async Read] Starts the asynchronous read process. At this time, check on “Subscribe”.
[Sync Write] Starts the synchronous write process.
[Async Write] Starts the asynchronous write process.
This is a kind of the asynchronous read process. When using “Subscribe”, the obtained data is sent to the client after it has been compared with the cache value. When clicking “Refresh”, all the cache values are cleared. Therefore, all values are always updated.

* Regarding the difference between the synchronous read process and synchronous read process, when accessing from the client (when calling up the function of the sever) in the synchronous read process, other process cannot be started until the server process is completed.
For example, when the server waits for the communication time-out time if the recorder is not powered ON, the client cannot start other process during this period.
On the other hand, when accessing from the client (when calling up the function of the sever) in the asynchronous read process, the server immediately returns the answer to the client. Then, after the process in the server has been completed, the server sends the process completion notification to the client. Therefore, the client can start other process without waiting for the process on the server.

Next, the subscribe means that the server does not return the data to the client if the difference between the previous data and current data does not exceed the specified level.
This may reduce the data send volume since data variations caused by noise are not notified to the client.
The constant value is called “DeadBand”, which is specified as % value to the span value of each item.

When writing the data, the following dialog box will appear.

Input a value in “Data” and specify the data type of the data you wish to send in “Type”. Click the “OK” button to send the data.
[Spreadsheet] Shows the statuses of the item IDs registered in the group.

- [ItemID] Shows the registered item ID.
- [Value] Shows the value of the item ID.
- [Time] Shows the time when the value of the item ID is obtained.
- [Quality] Shows the quality of the value of the item ID. For details about quality values, see IM.

In OPC, “Value”, “Time”, and “Quality” are sent to the client as a set when obtaining the item ID data. For details, see the reference about OPC Foundation.

- [TYPE (ans)] Shows the data type of the obtained value of the item ID.
- [TYPE (req)] Shows the data type of the item ID specified by the client.
- [TYPE (svr)] Shows the canonical data type of the item ID on the server.

[Error Information frame] Shows the error information.

- [Error Code] Set the error code output from the server.
- [Error String] Displays the contents of the specified error code. Basically, the contents are written in English.

Only errors output from the OS are displayed in the language used for the OS.

- [Server shutdown reason] Shows no data.
4.2 Utilizing the DAQOPC Client Tool Efficiently
By utilizing the DAQOPC Client Tool efficiently, the user can decide the trouble location about the connection between the user’s OPC Client and DAQOPC.

4.3 Exchanging the Data between OPC Client and DX/MV/DARWIN
The following shows the data exchange flow between the OPC Client and DX/MV/DARWIN.

1. The client software accesses DAQOPC through the OPC interface.
2. DAQOPC accesses DX/MV/DARWIN using the protocol of each recorder.
3. DX/MV/DARWIN returns the data in the DX/MV/DARWIN format to DAQOPC.
4. DAQOPC returns the data to the client software through the OPC interface.

4.4 Trouble Location
If DX/MV/DARWIN data cannot be obtained smoothly from the client software, the cause of this trouble is either the OPC interface or the part between DAQOPC and recorder. To find the cause, check the status between DAQOPC and DX/MV/DARWIN, and then check the OPC interface.

4.5 Trouble between DAQOPC and DX/MV/DARWIN
To find the cause of the trouble between DAQOPC and DX/MV/DARWIN, use the simulation function of DAQOPC. When using the simulation function of DAQOPC, DAQOPC does not communicate with the recorder, but it sends the virtual data to the client. To use the simulation function of DAQOPC, start up the DAQOPC setup software and select [Operation Setup] from the [Setup] menu to open the “Operation Setup” dialog box.
In this dialog box, check on “Run in demo mode.” and save the settings. After that, restart the OS. DAQOPC then becomes a simulator.

The following shows the data exchange flow when DAQOPC is operated as a simulator.

![Diagram of data exchange flow]

In this status, access DAQOPC using the OPC Client Tool. When the data can be obtained at this time, turn OFF the simulator function and restart the OS. Access DAQOPC again using the OPC Client Tool. At this time, if the access to the data is failed, the following may be the cause.

- The IP address of the unit specified by DAQOPC is incorrect. This can be checked using Ping.
  CAUTION ! At this time, run Ping in both power ON and OFF states. It is accepted that Ping runs successfully with the unit powered ON. However, if Ping runs successfully with the unit powered OFF, the specified address is used for a unit other than that specified.
- The password of the unit is incorrect. (For DX only)
4.6 Trouble in the OPC Interface

When the access to DAQOPC is succeeded using the OPC Client Tool, it is said that the OPC interface functions correctly. Therefore, if the access to DAQOPC is still failed using other company’s client software in this state, there may be a high possibility that the settings on other company’s software are incorrect.

If this happens, check the settings of other company’s client software. At this time, operate DAQOPC in the simulation mode to eliminate the effects of DXMV/DARWIN. If DAQOPC cannot be seen from other company’s client software when accessing DAQOPC from other company’s client software in the remote connection mode, it is thought that the client setup of DAQOPC is not performed. Therefore, check this point.

**CAUTION !**  When using Intouch, DAQOPC on the remote PC can be checked. However, if the client setup is not performed, the actual access cannot be performed. Carefully check this point.
5. Creating an Application Program Using Excel

Creating a VBA Macro

1. Launch Excel and create a new Book.
(3) Select [Tool] - [References] from the Visual Basic Editor menu. [References] shows the library files which can be referenced to. Select the Yokogawa Exaopc Data Access Automation Server.

(4) Select [Insert] - [Standard Module] from the Visual Basic Editor menu. A "Module" folder will be added in the project tree area; a "Module 1" item will be added under it.
(5) Describe the variable declare statement in the Module 1 (code) area.

(6) Select [Insert] - [Procedure] from the Visual Basic Editor menu. The "Add Procedure" dialog box will appear. Enter "Opc_Open" in the name field and click OK.

(7) A public subprocedure "Opc_Open" is added in the Module 1 (code) area.
(8) Describe the following code in "Opc_Open".

(Opc_Open procedure)

```vba
Public Sub Opc_Open()
    Dim strItemIDs(4) As String
    Dim lClientHandles(4) As Long
    Dim lErrors() As Long
    Dim i As Integer
    If objServer Is Nothing Then
        'Create the OPC server object.
        Set objServer = New OPCServer
        End If
    If objServer.ServerState = OPCDisconnected Then
        'Connect to the OPC server.
        objServer.Connect("Yokogawa.ExaopcDADAQCDXMV1")
        End If
    If objGroups Is Nothing Then
        'Create the OPC group collection.
        Set objGroups = objServer.OPCGroups
        End If
    If objTestGrp Is Nothing Then
        'Add a group.
        Set objTestGrp = objGroups.Add("Test")
        End If
    If objItems Is Nothing Then
        Set objItems = objTestGrp.OPCItems
        With Worksheets("Sheet1")
            For i = 1 To 4
                'Obtain the item name from the cell.
                strItemIDs(i) = .Cells(1, i).Text
                lClientHandles(i) = i
            Next i
            .Range("A2:D2").ClearContents
        End With
        'Add an item.
        Call objItems.AddItems(4, strItemIDs, lClientHandles, _
            lServerHandles, lErrors)
        End If
    End Sub
```

(9) Create the "Opc_Close" and "Opc_Read" procedures in the same manner.

(Opc_Close procedure)

```vba
Public Sub Opc_Close()
    Dim lErrors() As Long
    If objServer Is Nothing Then
        Exit Sub
    End If
    If Not objGroups Is Nothing Then
        If objGroups.Count > 0 Then
            'Delete the item.
            Call objGroups.Remove(4, lServerHandles, lErrors)
        End If
        Set objGroups = Nothing
        End If
    If objTestGrp Is Nothing Then
        'Delete the group.
        objGroups.Remove("Test")
        Set objTestGrp = Nothing
        End If
    If objItems Is Nothing Then
        Set objItems = Nothing
        End If
    If objServer.ServerState <> OPCDisconnected Then
        'Disconnect the server.
        Call objServer.Disconnect
        Set objServer = Nothing
        End If
    End Sub
```

(Opc_Read procedure)

```vba
Public Sub Opc_Read()
    Dim ItemVal() As Variant
    Dim lErrors() As Long
    Dim i As Integer
    If objServer Is Nothing Then
        Exit Sub
    End If
    If objServer.ServerState = OPCRunning Then
        'Read in.
        Call objTestGrp.SyncRead(OPCCache, 4, lServerHandles, _
            ItemVal, lErrors)
        With Worksheets("Sheet1")
            For i = 1 To 4
                'Set data in the cell.
                .Cells(2, i).Value = ItemVal(i)
            Next i
        End With
    End If
    End Sub
```

(10) Select [Debug] - [Compile VBA Project] from the Visual Basic Editor menu and check to make sure that there is no syntax error.
(1) Describe the Item IDs to be referenced to during the “Opc_Open” procedure in the cells A1 to D1 on an Excel sheet. Create a graph as necessary. Then create a button. To create a button, select [Show] - [Control Box] and the control box will appear. Select the “Button” control in the control box and place a button on the sheet as desired.

(2) Right-click the button placed on the sheet in Step (1) to call up the corresponding pull-down menu. Select “Property” from the menu to show the “Property” dialog box and change the Caption type to "Connect".
(3) Right-click the button placed on the sheet in Step (1) to call up the corresponding pull-down menu and select “Show Code” from the menu. The Visual Basic Editor will start and the call back procedure “Command Button 1_Click” when the command button is clicked will appear. Describe the Opc_Open procedure call statement in the procedure.

(4) In the same manner, set the "Disconnect" and "Read" buttons and describe the corresponding Opc_Open procedure call statements in the procedure.

(5) Clicking the “Connect” button will clear the values in the cells A2 - D2. Clicking the "Read" button will show the Item ID values in the cells A2 - D2. At the same time, the graph will be updated according to the values in the cells A2 - D2.