# Release Notes for DriveOPC 2.07

## Contents



**Note!** *This Release Note is distributed as a PDF-file (ReleaseNotes.pdf), which can be read by Acrobat Reader.*

# Introduction

This document contains release information about DriveOPC version 2.07, such as change history.

DriveOPC is an OPC Foundation Data Access Standard 1.0A compliant OPC Server for ACS600 drives.

# Compatibility

Since version 2.03 build 130, you can no more start using NISA-03 DDCS/ISA boards immediately after installing DriveOPC under Windows 2000 and Windows XP. You have to tell Windows their presence first (Add Hardware).

The older communication libraries do not work with the new drivers under Windows 2000 and Windows XP. The reason is that the drivers have been missing some parameter checks, which are now included, and the older communication libraries have a bug concerning the parameters (potentially crashing the operating system).

The following bug corrections may cause compatibility problems:

- Bug 2.00.100.09: Consistency of error code when quality is bad. Bad values read from a device do not return errors any more. The quality is just marked bad.
- Bug 2.00.100.17: Enumerated values not starting from one. The lower bound is not always zero any more but is the value corresponding the first string. Also, the value corresponding the first string was always one (other kinds of enumerations were not accepted). In the new release, the lower bound is one in this case.

# Enhancements

Following is a list of enhancements made.

## Enhancements in Version 2.07 Build 170

Compared with version 2.06 build 160.

**Additional features**

- Recognition of board types RMIO-1x has been added.

## Enhancements in Version 2.06 Build 160

Compared with version 2.05 build 150.

**Additional features**

- Support for DCS800 drive family has been added.
- Support for better drive clock synchronization has been added.

**The following item tags have been added:**

| | |
|---|---|
| `AllowZombies` | TRUE allows adding of items for non-identified drives. |

## Enhancements in Version 2.05 Build 150

There are no enhancements compared with version 2.04 build 140.

## Enhancements in Version 2.04 Build 140

Compared with version 2.03 build 130.

**Additional features**

- It is possible to introduce new kinds of drives in an INI-file.
- DriveOPC is added into component category “OPC Data Access Servers Version 1.0”. The category is also added in case it is not already in the registry.

## Enhancements in Version 2.03 Build 130

Compared with version 2.02 build 120.

**Additional features**

- Plug and Play compatible driver for NISA-03 DDCS/ISA board is included. It works in Windows 2000 and Windows XP operating systems. It means that a board has to be introduced to Windows (Add Hardware) before it can be used. It also means that Windows plug and play system can now handle resource conflicts concerning the board.
- Plug and Play driver for NDPA-02 DDCS/PCMCIA board now supports more than one board.
- Plug and Play driver for NDPA-02 DDCS/PCMCIA board does not request use of an interrupt line for a board any more.
- Maximum number of channels supported is 5 under Windows NT (1 NDPA-02 DDCS/PCMCIA board and 2 NISA-03 DDCS/ISA boards).
- Maximum number of channels supported is 10 under Windows 2000 and Windows XP. Maximum 2 NISA-03 DDCS/ISA boards are supported. Maximum number of NDPA-02 DDCS/PCMCIA boards under Windows 2000 and Windows XP depends on number of NISA-03 boards installed. With no NISA-03 boards, maximum 10 NDPA-02 boards are supported. With 2 NISA-03 boards installed, maximum 6 NDPA-02 boards are supported.
- Detection of the presence of a NISA-03 DDCS/ISA board has been enhanced. However, it is important under Windows NT only, because under Windows 2000 and Windows XP the plug

and play system prevents resource conflicts. Under Windows NT the better detection helps to avoid blue screens in case some other board is present at I/O addresses reserved for NISA-03 DDCS/ISA boards.

- Communication library now displays a message box in case it detects that hardware of a communication board fails. So called low level communication has also been enhanced in such a case. The failure is no more reported blindly by a call-back to the client. Instead, it is checked that the client has configured the call-back functions (Drive*Debug* does not crash any more, for example).
- New version of NISADUMP.EXE (version 2.0) is included. It knows how to interact with the new drivers. The old NISADUMP.EXE does not work under Windows 2000 and Windows XP any more.
- Detecting of presence of drives is now somewhat faster, especially in case the number of drives is small.
- Under Windows 2000 and Windows XP, power saving of the PC is automatically inhibited, if the operating system and hardware support the inhibition. If not, or the operating system is Windows NT, power saving must be inhibited manually. Note that power saving is inhibited automatically only while the application using DriveOPC is running.

## Enhancements in Version 2.02 Build 120

Compared with version 2.01 build 110.

### Additional features

- Plug and Play driver for NDPA-02 DDCS/PCMCIA board is included. It works in Windows 2000 and Windows XP operating systems.
- Communication library now displays a message box in case another application already uses the library (even by another client session). The message is not shown in case DriveOPC is configured to be used remotely.

## Enhancements in Version 2.01 Build 110

Compared with version 2.0 build 100.

### Additional features

- First time checking the presence of a drive (identifying) has been enhanced.
- Vendor info now shows also information about the DriveOPC.
- Time stamp accuracy has been increased about five orders of magnitude, if the PC hardware supports the performance counter.
- Memory leaks in some rare exception handling have been removed.
- Support for fast mode monitoring functions now properly and is synchronised with normal item handling.
- Uploading an empty datalogger is now handled properly.

### Item tags removed

Item `{Ch}{Node}DL[1|2].Status.Empty` has been removed.

**The following item tags have been added:**

- Communication:

| | |
|---|---|
| `Communication.Restart` (write only) | TRUE restarts the communication. |
| `Communication.NoReplyToDownDelay` | Time in milliseconds, after which a drive is considered to be down, if there is no answer. This item is used unless the corresponding item for a drive overrides it. A negative value means that drives are never considered to be down. Default value is 5000. |
| `Communication.NoReplyRetryInterval` | Time in milliseconds, which defines the interval used to probe a drive that is down. This item is used unless the corresponding item for a drive overrides it. Default value is 2000. |
| `{Ch}{Node}Communication.NoReplyToDownDelay` | Time in milliseconds, after which the drive is considered to be down, if there is no answer. Unless negative, this item overrides the corresponding common item. Default value is -1. |
| `{Ch}{Node}Communication.NoReplyRetryInterval` | Time in milliseconds, which defines the interval used to probe a drive that is down. Unless negative, this item overrides the corresponding common item. Default value is -1. |
| `Communication.ConfigureDialog` (write only) | The number of the DDCS communication channel, which is to be configured. Configuring is done by displaying a dialog box. Does nothing in case DriveOPC is configured to be a remote server. |

- Status (DCS600, read-only):

| | |
|---|---|
| `{Ch}{Node}Status.Ready` | TRUE indicates that the DCS600 drive is ready. |

- Datalogger status (read-only):

| | |
|---|---|
| `{Ch}{Node}DL[1\|2].Status.Trigged.Difference` | TRUE indicates logger trigged to a difference. |

- Common data logger properties (read-only):

| | |
|---|---|
| {Ch}{Node}Properties.DLDescription.Loggers | Number of dataloggers. |
| {Ch}{Node}Properties.DLDescription.TimeLevel | Logging time level in microseconds. |
| {Ch}{Node}Properties.DLDescription.Samples | Maximum number of samples. |
| {Ch}{Node}Properties.DLDescription.Channels | Number of channels. |

- Available datalogger features (read-only):

| | |
|---|---|
| {Ch}{Node}Properties.DLDescription.Available.RealTime | Is real time available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerFault | Is fault triggering available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerLevel | Is level triggering available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerLimit | Is limit triggering available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerWarning | Is alarm triggering available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerUser | Is user triggering available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerDifference | Is difference triggering available? |
| {Ch}{Node}Properties.DLDescription.Available.TriggerExternal | Is external triggering available? |

- Datalogger settings:

| | |
|---|---|
| {Ch}{Node}DL[1\|2].Settings.Trig.User | TRUE indicates that the logger must trigger by a user command. |
| {Ch}{Node}DL[1\|2].Settings.Trig.Difference | TRUE indicates that the logger must trigger when a difference occurs. |
| {Ch}{Node}DL[1\|2].Settings.Trig.External | TRUE indicates that the logger must trigger when an triggering occurs. |
| {Ch}{Node}DL[1\|2].Settings.Samples (read-only) | Size of a channel buffer (you cannot always rely on this information). |

- Parameter and signal descriptions (read-only):

| | |
|---|---|
| {Ch}{Node}Par.m.Description | Description of group m. |
| {Ch}{Node}Par.m.n.Description | Description of parameter or signal n in group m. |

- Kind of drive (read-only):

| | |
|---|---|
| {Ch}{Node}Properties.Kind | Kind of the drive (empty or ACC600, ACF600, ACN600, ACP600, ACS1000, ACS600, ACS6000C, ACS6000C-CC, ACS6000SD, ACS6000SD-FE, ACW600, DCS600, NCB, NTY, ACC800, ACF800, ASN800, ACP800, ACS800, ACW800). |

**New Examples Added**

The following new examples have been added:

- HelloLogger
- Three scripting versions of HelloOPC (vbs, VBA, IE)

# Bugs Corrected

Following is a list of corrected bugs found in previous versions.

## Bugs Corrected in DriveOPC Version 2.07 Build 170

No bugs in DriveOPC version 2.06 Build 160 have been reported. So, there are no new bug corrections.

## Bugs Corrected in DriveOPC Version 2.06 Build 160

The bugs were in DriveOPC versions 2.05 Build 150, 2.04 Build 140, 2.03 Build 130, 2.02 Build 120, 2.01 Build 110, and 2.0 Build 100.

**2.05.150.01: Incorrect timing of active groups**

The cycle times of active groups were handled incorrectly in DriveOPC. The actual cycle times were most of the time shorter than specified by the client, and they did not stay constant.

**2.05.150.02: Corrupted .X01 or .CLD file may cause crashing**

There is not much checking done in DriveOPC when reading a .X01 or .CLD file. Reading of a corrupted file may cause an exception to be thrown.

DriveOPC was not prepared to catch such an exception, which could cause crashing of the server (SMP.EXE) or the client (SMP.DLL).

## Bugs Corrected in DriveOPC Version 2.05 Build 150

The bugs were in DriveOPC versions 2.04 Build 140, 2.03 Build 130, 2.02 Build 120, 2.01 Build 110, and 2.0 Build 100.

**2.04.140.01: Too short time-out in FLASH compression**

Some drive FLASH PROM operations executed within DriveOPC request in some rare cases compression of the FLASH . However, the time-out in the internal compression is set too short (for at least drives with big FLASH PROMs). Thus the operations fail.

The bug has been corrected. The correction removes the internal DriveOPC compression requests totally, which means that the client has to ask FLASH compression if it is required. An item has been added for such requests. Also, the client has to give the time-out value.

**2.04.140.02: Non-real triggering of a datalogger is not handled correctly**

Non-real triggering variables, triggering levels, and hysteresis of a datalogger are not handled correctly. Everything is converted into real, which is against DDCS specification.

The bug has been corrected.

**2.04.140.03: Overflow in calculating time stamps from high performance counter**

There is an overflow in the formula, which calculates time stamps from the high performance counter.

Symptoms: If DriveOPC has been running several days in a PC containing a high performance counter, time stamps suddenly jump backwards and active groups stop cycling.

Note that all modern PCs have a high performance counter.

The exact run time before the error happens varies from PC to PC, because it depends on the frequency of the counter.

The bug has been corrected.

## Bugs Corrected in DriveOPC Version 2.04 Build 140

The bugs were in DriveOPC versions 2.03 Build 130, 2.02 Build 120, 2.01 Build 110, and 2.0 Build 100.

#### 2.03.130.01: Browsing is inconsistent with item validation

Especially in case of a positional drive, browsing included some Status and Control items, which do not exist (do not validate when items are added), and did not contain all Status and Control items.

The bug has been corrected.

#### 2.03.130.02: Timeout in writing an OS Command is not handled properly

Symptoms: Clearing FPROM fails.

If timeout occurs in writing an OS Command, recovery is not made correctly in DriveOPC. Some drives may go into a state, where the drive does not accept a FPROM formatting command until it has been restarted.

The bug has been corrected.

#### 2.03.130.03: An Item ID has been attached to a branch

Item ID {m}{n}DL[1|2].Trigged was attached directly to the branch ...->Data logger[ 1| 2]->Status instead of being a leaf under the branch.

Although DriveWindow could handle such kind of an error, many client programs were not able to find the item while browsing. So, they did not show Trigged as an item under the Status branch.

The bug has been corrected.

## Bugs Corrected in DriveOPC Version 2.03 Build 130

The bugs were in DriveOPC versions 2.02 Build 120, 2.01 Build 110, and 2.0 Build 100.

#### 2.02.120.01: ACS PMM and ACN PMM are not recognised

ACS 600 Single and MultiDrive for Permanent Magnet Synchronous Motors (ACS PMM and ACN PMM) are now recognized by DriveOPC. Previously, value of Properties.Kind was ACS600.

#### 2.02.120.02: Disconnect does not release the synchronisation lock under Win2000/XP

Symptoms: Under Win2000 or WinXP, a client program, which is using the in-process OPC server (SMP.DLL), disconnects the server, but continues to run. However, communication library stays locked to the client program, and no other program is able to use the library. Note that under WinNT the lock is released properly.

Cause: Disconnecting the in-process server does not free the DLL. When the server is disconnected, it tries to free the communication library. However, there is a book keeping error in the server, which affects the unloading of the communication library under Win2000/XP. So, the global semaphore, which is handled during loading and unloading of the communication library, stays locked to the client application as long as it runs.

The bug has been corrected.

#### 2.02.120.03: Upper 32 bits of time stamps in fault logger are lost

Symptoms: Fault logger time stamps "overflow" after about 120 hours and differ from the value shown by the operator's panel.

Cause: Time stamps of faults (and events) are uploaded as 64 bit integers (unit is 100 microseconds). When a time stamp is "time since last power-up", DriveOPC loses the high 32 bits when converting the time stamp to a string.

The bug has been corrected.

## Bugs Corrected in DriveOPC Version 2.02 Build 120

The bugs were in DriveOPC versions 2.01 Build 110 and 2.0 Build 100.

### 2.01.110.01: Reading Par.m.n.Description from device may fail

Reading Par.m.n.Description from device fails in case the parameter has no unit.

The bug has been corrected and reading a description from device does not fail any more.

### 2.01.110.02: Power saving may considerably slow down communication

Symptoms: DDCS (S-protocol) communication is very slow unless the mouse is moved all the time or keys are repeatedly pressed.

In some computers (especially portables with ACPI compliant BIOS and Win2000 or WinXP) there is an automatic power saving mode, which stops the processor clock (ACPI state S1). Because the Pentium processor time stamp counter, which counts clock cycles, is used in detecting multimedia timer interrupt bursts, the detection fails (the counter does not run during ACPI state S1). The detection thinks that the multimedia timer is bursting all the time, and does not execute the S-protocol.

The bug has been corrected. Instead of the Pentium processor time stamp counter, the high-resolution performance counter is now used (if it exists).

### 2.01.110.03: No hardware access synchronization between simultaneous sessions

This bug concerns only Win2000 and WinXP, in which each client session (Windows station) has a kernel name space of its own. The internal synchronization mechanism within the communication library fails, because it references the (default) session name space instead of a new global kernel name space.

Symptoms: If DriveOPC is used both locally and remotely at the same time, both instances of DriveOPC have access to the drive communication hardware without knowing about each other. The result is that communication is seriously disturbed. In WinXP there can also be several local users logged on simultaneously, each being able to access the communication hardware, thus resulting communication disturbances.

The bug has been corrected. Also, the communication library has been enhanced to show a message box in case another application program is already using the communication hardware. The message box is not shown in case DriveOPC is configured to be used remotely.

### 2.01.110.04: Erroneous conversion of enumeration strings from OEM to ANSI

There is a bug in converting enumeration strings, which are in drives coded as (MS-DOS) OEM characters, to ANSI, which is used in Windows.

Symptoms: If language using letter “Ä”, for example, is selected, and an enumerated value in that language contains such a letter in other than the last value, the conversion fails. Instead, a garbage letter is show in such a position. Note that the last value is converted correctly.

The bug has been corrected.

Note that only language dependent strings (parameter and group names, parameter units, value enumeration strings, and event descriptions) are converted from OEM to ANSI. Parameter values, which are strings, are never converted. It means that such values are or should be ANSI coded also in drives.

## Bugs Corrected in DriveOPC Version 2.01 Build 110

The bugs were in DriveOPC version 2.0 Build 100.

### 2.00.100.01: IOPCGroupStateMgt::SetState may return S_FALSE

IOPCGroupStateMgt::SetState may occasionally return S_FALSE instead of S_OK.

S_FALSE is not listed by OPC Data Access Specification 1.0A in the allowed HRESULT return codes for the method.

S_FALSE is returned in case the client has not set a callback (pointer to IAdviseSink by calling IDataObject::DAdvise) for the group.

The bug has been corrected and S_FALSE is not returned any more.

**2.00.100.02: Client item handles must be unique**

Client side item handles (OPCHANDLDE) must be unique within a group.

OPC Data Access Specification 1.0A states that no assumptions should be made about client handles.

The methods, which have this requirement are IOPCItemMgt::AddItems and IOPCItemMgt::SetClientHandles.

The bug has been corrected and the client side handles need no more be unique.

**2.00.100.03: Avoid IOPCItemMgt::SetClientHandles**

Do no use IOPCItemMgt::SetClientHandles to make any changes into the client side item handles of a group.

If changes are made, anomalies in reading and writing may consequence, especially in asynchronous I/O and IAdviseSink callbacks. Also it is quite probable that DriveOPC crashes, latest when it is shutting down.

The bug has been corrected and the client side handles can be freely changed.

**2.00.100.04: Memory leak in IAdviseSink::OnDataChange callback**

A global memory area allocated by DriveOPC before calling IAdviseSink::OnDataChange is not freed by DriveOPC as it should be.

It means that if the client has given to an IAdviseSink interface to DriveOPC by calling IDataObject::DAdvise, there is a cumulative memory leak. Eventually the program will slow down and may crash because running out of memory.

The bug has been corrected and there is no memory leak any more.

**2.00.100.05: Recovery after drive power-up**

If a drive is powered down and then up again while DriveOPC is running, communication with the drive is not recovered.

Note that automatic drive power-up recovery will never be foolproof. Especially, when node number 1 is in use or several drives are powering up at the same time, there is a risk not to be able to recover automatically.

The bug has been corrected and the communication recovers after power-up (or communication link failure).

**2.00.100.06: Recovery after branching unit power-up**

If the DDCS network contains a branching unit, which requires programming after power-up, communication through the branching unit is not recovered, when the branching unit is powered down and then up again while DriveOPC is running.

There is no automatic recovery, but a new item, which allows the client program to request restarting of the communication. Restarting of the communication reprograms the branching units.

The bug has been corrected. Correction of the bug does not do the recovery automatically. The reason is that it is not possible to detect powering-up or down of the branching unit by the DriveOPC. The correction requires TRUE to be written by the client into the write-only item "Communication.Restart" (Note that there is no channel and no node). The item is not shown in browsing. All branching units found are reprogrammed and it is a lengthy operation.

**2.00.100.07: Configuring of a datalogger**

Configuring of a datalogger does not work properly. Actually, only changing of configuration values of a properly configured datalogger works.

How to configure a datalogger depends on the AMC-board datalogger implementation. The following rules are usual, partly implemented in DriveOPC:

- A datalogger can be configured only if it is not running.
- There are restrictions in the order of configuration. For example, channel variables are required to be entered in increasing order. Clearing a channel variable will clear all channel variables with a bigger channel number.
- If there is no triggering variable, triggering level and hysteresis will be uncertain zeros. Entering then a triggering variable may clear triggering level and hysteresis from the AMC-board. Thus triggering configuration order can be critical, too.

The bug has been corrected.

**2.00.100.08: Accessing single datalogger numbered 0**

DriveOPC is not able to access a single datalogger on some boards. Some system software versions have numbered the single datalogger to be number zero. DriveOPC is able to access a single datalogger only, if it is numbered one. Symptom is that reading any value (even any of the status values) from the datalogger returns quality QUAL_BAD.

The bug has been corrected. The new release checks and uses the numbering used by system software in case of a single datalogger.

**2.00.100.09: Consistency of error code when quality is bad**

DriveOPC returns error code E_FAIL about a value, quality of which is QUAL_BAD and which is read from the device. If the value is read from cache, no error (S_OK) is indicated if quality is QUAL_BAD. This means that there is inconsistency in handling of QUAL_BAD values.

The bug has been corrected. The new release returns no error code (S_OK) whether a QUAL_BAD value is read from cache or device.

Note that correction of the bug may require changes to be made into the client program because bad values read from a device do not return errors any more. The quality is just marked bad.

**2.00.100.10: Data type of bad quality may not be consistent.**

When reading a value from cache while the quality of it is QUAL_BAD, DriveOPC may occasionally return data type other than VT_EMPTY.

The bug has been corrected. The new release returns always values of QUAL_BAD quality with data type VT_EMPTY.

**2.00.100.11: Datalogger may contain spurious data**

Currently number of elements returned by DriveOPC from a datalogger channel is always the full size of the logger channel. However, it may be that the drive in some cases gives fewer values than is the datalogger size. In such a case the rest of the values given to the client will be random.

The bug has been corrected. The new release returns always the correct number of elements..

**2.00.100.12: Datalogger quality and time stamp may not be correct**

DriveOPC has internally a common quality and time stamp for all data channels. When reading from cache, the quality and time stamp are thus actually from the last reading of any of the channels from the device, which may not be the one that the client is reading.

The bug has been corrected. In the new release there is internally quality and time stamp separate for each channel.

**2.00.100.13: Communication overload when drive or communication is down**

If, for some reason, communication with a drive stops working (drive is down or optical fiber is disconnected), there is a high probability of communication overload. The reason is that each query that DriveOPC makes from such a drive goes to time-out. DriveOPC also does some retries, all of

which go to time-out as well. Thus the communication slows down considerably. Some measurements show that a query, which normally takes 4 ms, takes 160 ms in case the drive does not response.

For example, having an active group containing 500 active items with 4 second update rate will normally cause about 50 % communication load. If the optical fiber is disconnected, reading will take about 80 seconds (2000 % communication load). Because active groups are handled internally in DriveOPC by high priority threads and all the data, even the data in cache, is locked during reading of a group, a client trying to read from cache hangs.

The new release of DriveOPC detects and handles this situation better.

Correction of the bug introduces two new items at common level (no address before the item) and drive level (channel and node in curly braces precede the item). If a drive specific item value is negative, the corresponding common item value is used for the drive, otherwise the drive specific value is used.

The new items are:

| | |
|---|---|
| [{Ch}{Node}]Communication.NoReplyToDownDelay | Time in milliseconds, after which a drive is considered to be down, if there is no answer. |
| [{Ch}{Node}]Communication.NoReplyRetryInterval | Time in milliseconds, which defines the interval used to probe a drive that is down. |

Default common values are 5000 (delay) and 2000 (interval). Default drive specific values are -1 (use the common values).

Note that when DriveOPC considers a drive to be down, it stops trying to read values from the drive. However, from time to time, a read request probes the drive to see if it has come up again.

**2.00.100.14: Slow start-up if drive is down.**

DriveOPC may start-up slowly in case the first drive(s), items of which the client program tries to add or validate, is down or its fiber optic link is broken. The reason is that if there is not yet any drive successfully connected, DriveOPC closes the communication libraries after a failed connection try. This means that the libraries are opened and closed for each failed try until a successful one is detected.

The bug has been corrected. The libraries are not closed after a failed try.

**2.00.100.15: Browsing does not detect pass-code changes**

When the client program first time browses down (by IOPCBrowseServerAddressSpace::ChangeBrowsePosition) into "Parameters", only those groups that are not currently protected by pass-code are taken into account. During new browsing the pass-code protection is not checked any more and the groups remain the same, even if pass-code has been changed. So, pass-code changes after the first browse have no effect to browsing.

The new version re-checks the protection.

**2.00.100.16: In-process DriveOPC (SMP.DLL) does not close**

Even when all references to the in-process DriveOPC objects have been released, SMP.DLL remains loaded (at least 6-12 minutes). It means that reconnecting while it is still loaded happens quickly, but no re-identification of the DDCS network is done.

This is actually a feature of Microsoft implementation of singleton object. The correction will mean that the in-process version of DriveOPC will no more be a singleton. Thus each CoCreateInstance will create a new in-process DriveOPC object. Although all precautions will be made to get the objects co-work, we recommend that any client program creates only one instance of a DriveOPC object.

In the new version the DriveOPC is no more a singleton and the in-process DriveOPC closes, when all objects have been released.

### 2.00.100.17: Enumerated values not starting from one

DriveOPC rejects enumerated values, which do not start from one. Such values have no EUInfo available in OPCITEMATTRIBUTES.

The bug is corrected in the new release. The correction is such that the lower bound of the SAFEARRAY in vEUInfo of OPCITEMATTRIBUTES will be the enumeration starting value.

Note that there are also VT_BOOL enumerated values (lower bound will be 0). The actual Boolean value FALSE will correspond to SAFEARRAY element 0 and TRUE SAFEARRAY element 1.

Note also that there may be compatibility problems in client programs, because previously the lower bound was zero although the starting value was one. In the new release the lower bound will be one.

### 2.00.100.18: OPCITEMATTRIBUTES erroneously reports dwEUType analog

Even if the drive has no minimum and maximum value for a parameter, DriveOPC erroneously says in OPCITEMATTRIBUTES that its dwEUType is analog (1). The minimum and maximum values are zero.

Note that even string parameters are reported to be of type analog with minimum and maximum values.

The bug has been corrected.

Note that a parameter of canonical type VT_BOOL can be reported to be of dwEUType analog if the drive says that it is not enumerated and has minimum and maximum value.

### 2.00.100.19: String values longer than 32 characters are not written correctly

Values of type so called visible string (canonical type is VT_BSTR), which are longer than 32 characters, are not written correctly to the drive. The characters after the 32nd one are garbage.

The bug is corrected in the new release. If the length is bigger than 255, only the first 255 characters are written by DriveOPC, however.

Note that the drive has a parameter dependent limit of the length and may do further cutting of the written value.

### 2.00.100.20: Precision lost when writing floating-point values

When values are written to a device, they are first converted to string internally by DriveOPC. The string is then converted to the canonical type. The standard conversion routine does not include enough digits to the string in case the value is floating-point (e.g., precision of six digits in case of VT_R4). Thus, occasionally the least significant digit may not be correct in case the canonical type is also floating-point.

The bug is corrected in the new release. The additional internal conversion to string has been removed, so clients using the canonical type when writing do not lose any precision any more.

### 2.00.100.21: Message about missing communication hardware repeats

When there is no drive communication hardware present in the PC, and DriveOPC has not been configured for remote use, the message "No Communication Hardware Found" repeats several hundred times.

The bug has been corrected. The message does not repeat, unless there is at least 3 seconds delay between consecutive communication library initializations caused by DriveOPC client calls.

### 2.00.100.22: Unsupported datatype conversion does not fail

According to OPC Specification, if an OPC Server is not able to convert value an item to the requested datatype, validation or adding such an item should fail. If the canonical datatype or the requested datatype is of type VT_ARRAY, DriveOPC may not be able to do a conversion. However, it may accept such a requested datatype, but returns the canonical datatype, when the item is read.

The bug is corrected in the new release. However, because standard conversion routines are used in converting the datatype when reading an item, a robust client should always check the datatype returned by the OPC Server, even if it has requested other than canonical datatype.

**2.00.100.23: Group destruction may crash DriveOPC**

When a group is destroyed in the DriveOPC, it may be that the internal thread servicing the group dies too late, i.e., it tries to reference the already destroyed group object, which crashes the application. Probability of such a situation increases, if there are several active groups with short interval and several active items.

In the new version the thread checks that after it has potentially slept, the group still exists.

**2.00.100.24: Fault logger shows wrong status of warnings**

Fault logger does not show the active status of warnings. Instead, it erroneously decodes and shows the acknowledgment status, as if the logger were an event logger.

The bug has been corrected.

**2.00.100.25: Client group handles must be unique and must not be changed.**

Client side group handles (OPCHANDLE) must be unique and they must not be changed.

OPC Data Access Specification 1.0A states that no assumptions should be made about client handles.

The methods, which have these requirements, are IOPCServer::AddGroup (unique handle) and IOPCGroupStateMgt::SetState (the handle must be the same as set by AddGroup).

The bug has been corrected. Client side group handles need no more be unique.

## Bugs in Examples of DriveOPC Version 2.0 Build 100

Corrected in version 2.01 build 110.

**The following corrections have been made:**

- In BrowseOPC.cpp added `break;` after line 513
  (`case VT_STORAGE: strCanonicalDataType += "VT_STORAGE";`).
- In ReadOPC.cpp added `break;` after line 396
  (`case VT_STORAGE: strCanonicalDataType += "VT_STORAGE";`).
- In WriteOPC.cpp added `break;` after line 383
  (`case VT_STORAGE: strCanonicalDataType += "VT_STORAGE";`).
- In COPCItem.h replaced line 271
  (`#define QUAL_SUBSTATUS_MASK 0x00C0`) with
  `#define QUAL_SUBSTATUS_MASK 0x003C`.
- In COPCItem.h added `#define QUAL_LIMIT_MASK 0x0003` after line 249
  (`#define QUAL_CONSTANT 0x0003`).
- In CCmdLine.cpp added the following after line 214 (}):
```
if (IsSwitch[*pCurrentChar]==false && *pCurrentChar &&
   m_SwitchCharacters.Find(*pCurrentChar)<0 && *pCurrentChar!='"' &&
   (!m_WhitespaceIsSeparator || !isspace(*pCurrentChar)) &&
   m_SeparatorCharacters.Find(*pCurrentChar)<0)
{  m_Reset = true;  // Error.
   break;
}
```
- Replace in BrowseOPC.cpp line 190, in ReadOPC.cpp line 190, and in WriteOPC.cpp line 170
  (`fprintf(stdout, "       Version = %u.%u Build %d.\n",`) with
  `fprintf(stdout, "       Version = %u.%02u Build %d.\n",`
- In case EUType of an item is OPC_ENUMERATED, the property Min of COPCItem in the wrapper classes now contains the value corresponding the first enumeration string.
- In ReadOPC.cpp, WriteOPC.cpp, and BrowseOPC.cpp, the property Min of COPCItem is used instead of 1 in case EUType of an item is OPC_ENUMERATED.