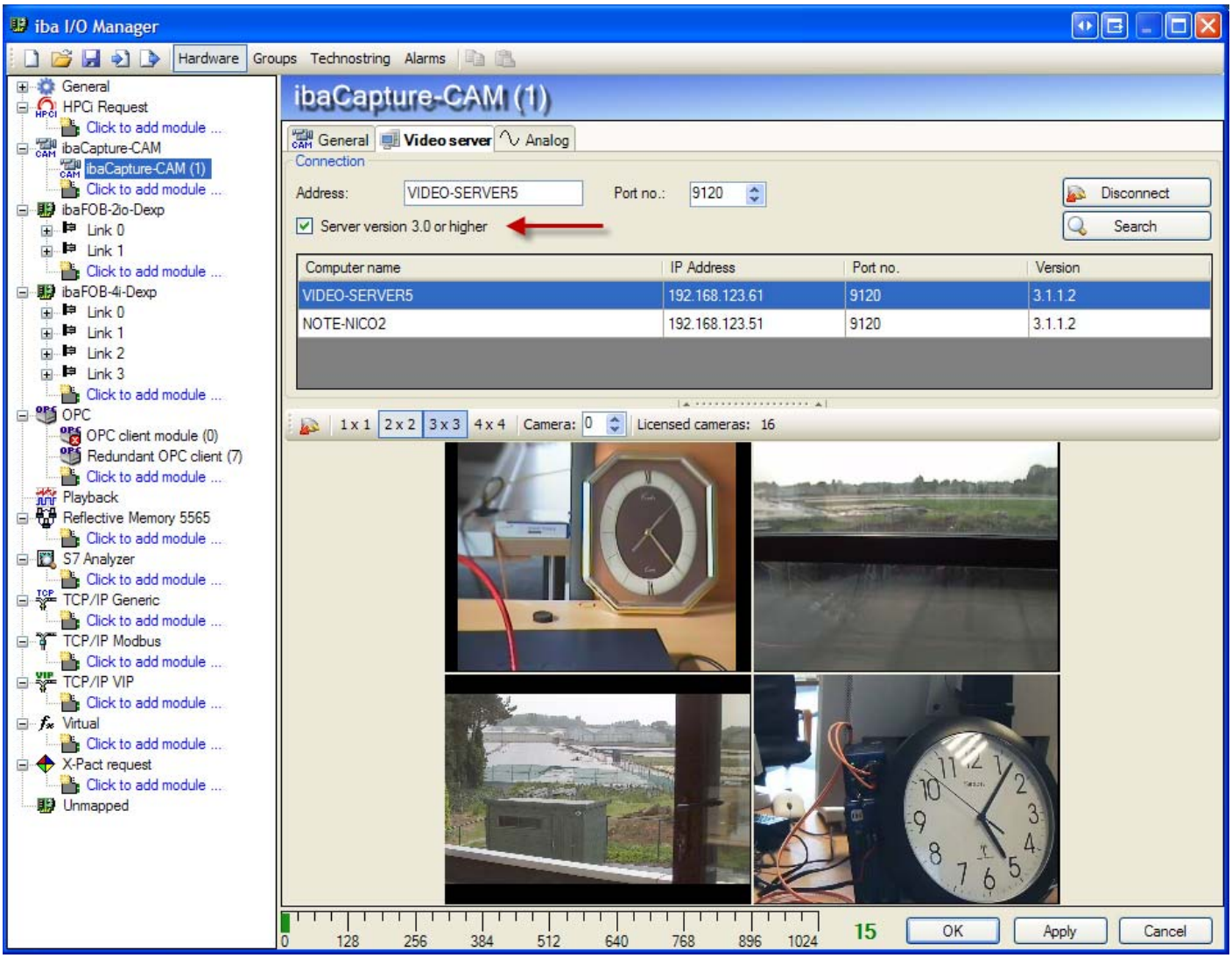
New features in ibaPDA v6.26.0

1 Iba-Capture-CAM v3.0

There is a new version of ibaCapture-CAM that is developed completely by iba. This new version is compatible with the video data generated by the previous version. ibaPDA can connect to both new and old versions of the ibaCapture-CAM server software. In the I/O manager there is a new checkbox on the connection tab of an ibaCapture-CAM module. This checkbox determines if you are working with a 3.0 server or an older one. The checkbox will be filled in automatically when you connect to a server. It will also be set when the acquisition is started in ibaPDA. This checkbox is only necessary when the ibaCapture-CAM server is not connected when the acquisition is started so that ibaPDA is not able to determine the version automatically.

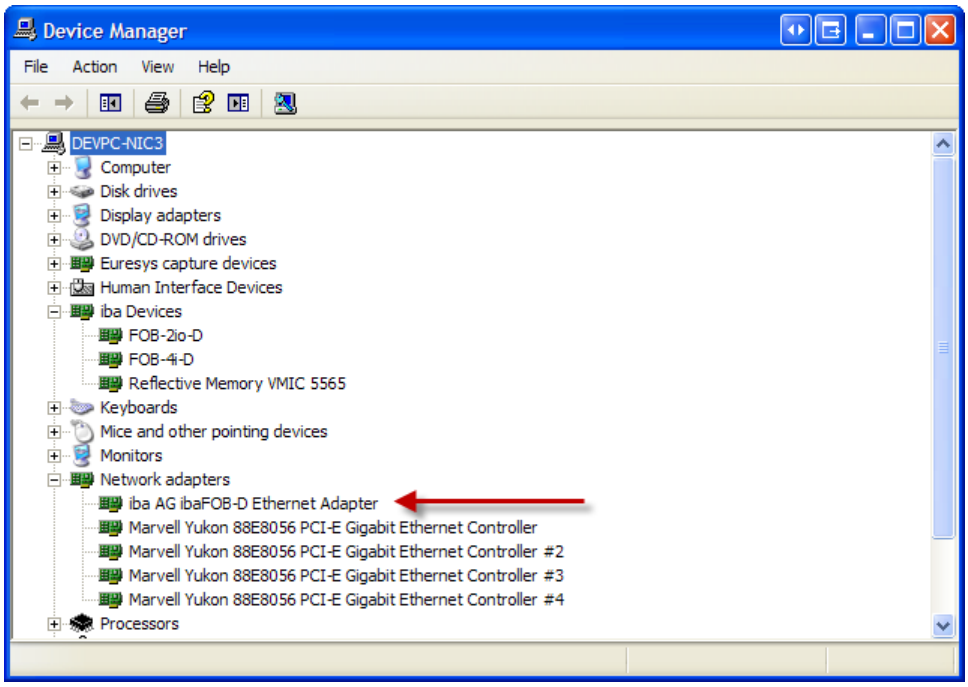


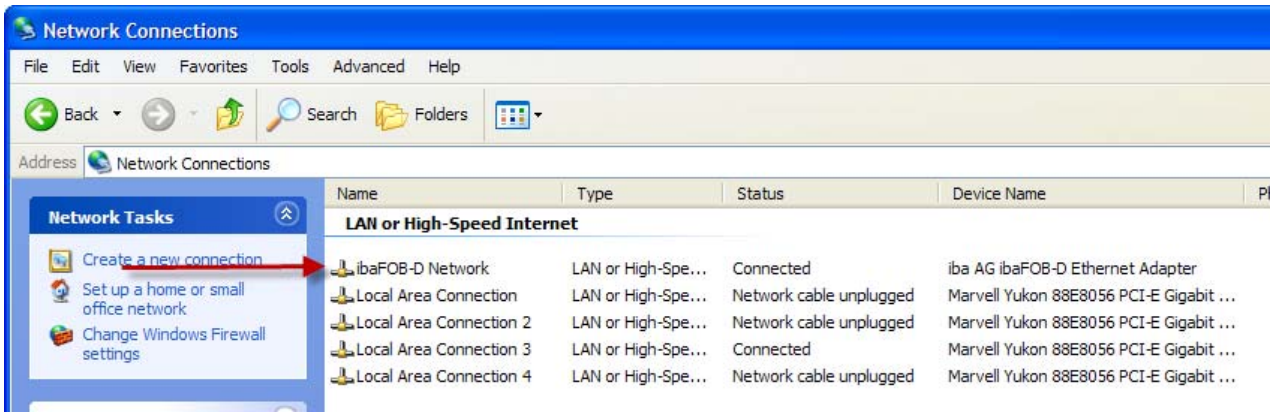
ibaPDA requires ibaCapture-CAM player v3.1 to be installed in order for it to display video. This player version is required even if the ibaCapture-CAM server is not a v3 server. The camera view can simultaneously show video coming from v3 and older video servers.

2 32Mbit Flex protocol

The 32Mbit flex protocol is a new protocol on an iba fiber optic link. It is supported by FOB-D boards with firmware C1 or higher and the ibaFOB-D-expresscard with firmware C1 or higher. The flex protocol is a 32Mbit protocol that supports up to 15 devices on 1 link. The sampling speed can be between 10 microseconds and 2000 microseconds. The FOB-D boards allow 2 time classes on a link but currently ibaPDA only supports 1 time class. One time class means that all devices on the link have the same speed. The link also supports a special channel that is used for Ethernet communication. This Ethernet channel can be used to send TCP/IP messages over the link in parallel with the fast acquisition data. The 32Mbit flex protocol can be run with 1 FO cable in open-chain mode and with 2 FO cables in ring mode. In open-chain mode only unidirectional communication from device to FOB-D board is possible. In ring mode bidirectional communication between device and FOB-D board is possible.

The ibaPDA 6.26.0 installer contains the ibaFOB-D network installer. This installer creates a new network card called the iba AG ibaFOB-D Ethernet Adapter and it creates a new network connection called ibaFOB-D Network.





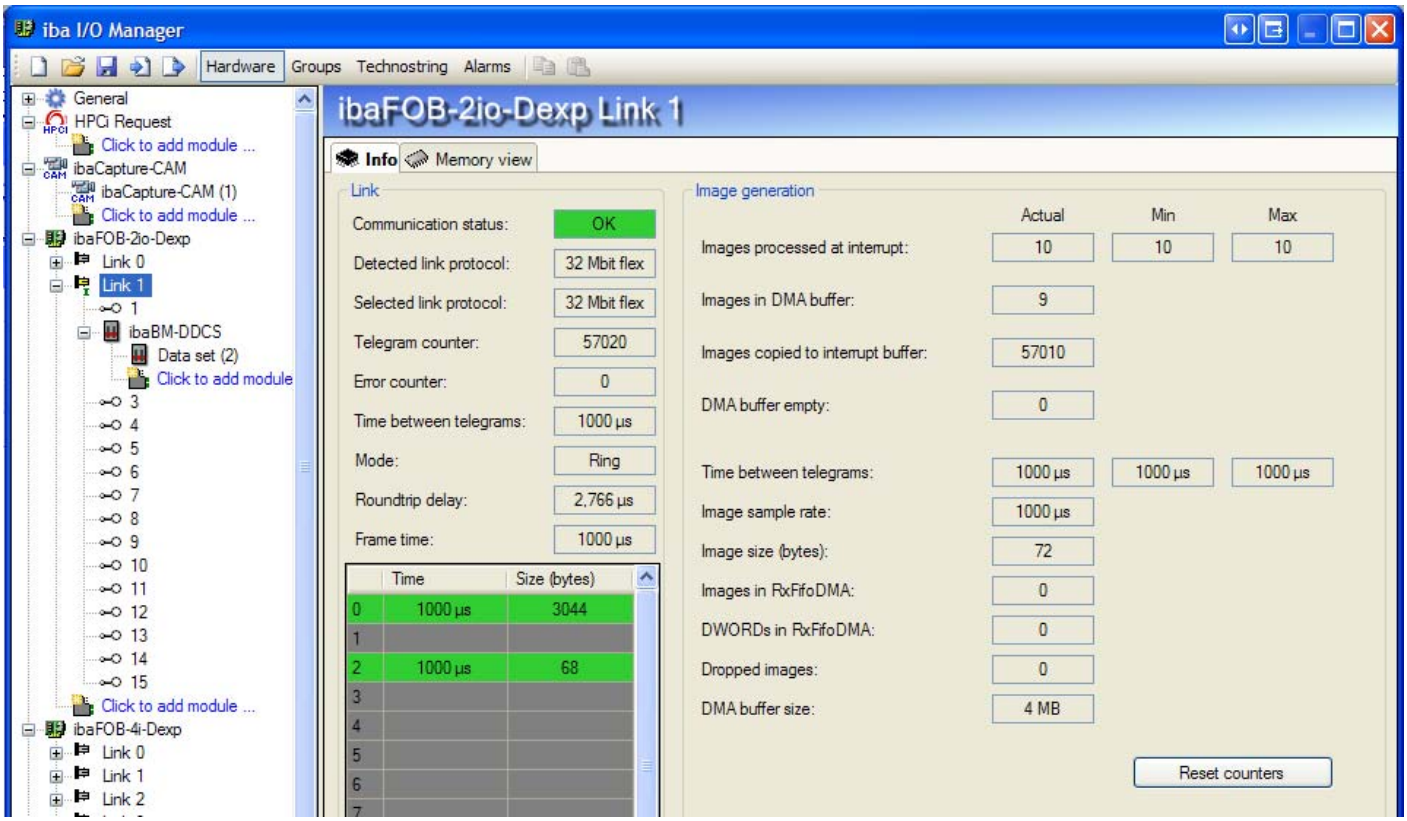
This network connection is used to communicate via TCP/IP with flex devices connected to the FOB-D boards in the system. There is only 1 network connection for all FOB-D boards. A Class B type IP address 172.16.0.100 with subnet mask 255.255.0.0 is assigned to the network connection by default via the installer. It is possible to change this IP address but currently ibaPDA only supports the default IP address.

The IP address of a flex device is determined by its connection to the system. The IP address has this form 172.16.x.y. X equals the board number times 10 plus the link number on the board. So if the device is connected to the first link of the first board then x = 0*10 + 0 = 0. If the device is connected to the third link of the second board then x = 1*10 + 2 = 12. Y equals 100 plus the value of the address hex switch on the flex device. So if you have 2 flex devices with hex switch set to 1 and 10 connected to the second link of the fourth board then the IP addresses will be 172.16.13.101 and 172.16.13.110.

You can find more details about the ibaFOB-D network installer and the operation of the ibaFOB-D network in the document: TechSpec_ibaFobDNetwork_v1.1.doc

At the moment there are 3 devices in ibaPDA that support the flex protocol:

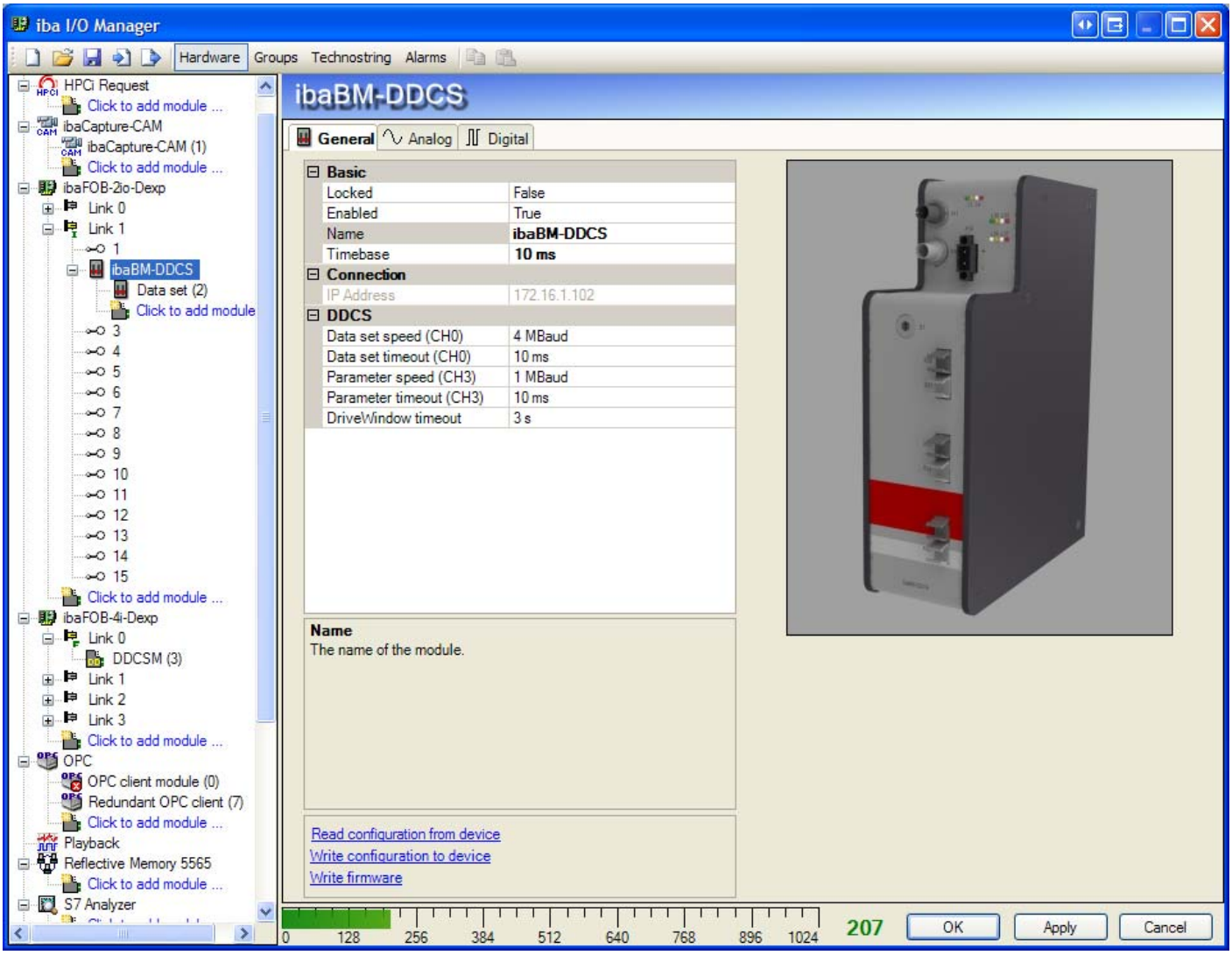
- ibaPADU-S-IT-16
- ibaPADU-S-CM
- ibaBM-DDCS



When a link is in flex mode then the diagnostics view on the links shows some interesting information about the communication. The mode can be either ring or open-chain. In ring mode the rountrip time is displayed. This roundtrip time will depend on the length of the cables and the number of connected devices. The frame time shows the configured time between flex frames. This frame time is the greatest common divider of the timebases of all devices on the link. It must be in the range 10 microseconds to 2000 microseconds. The time between telegrams is the time measured by the FOB-D board. It should be equal to the configured frame time. The table has 16 rows: row 0 for the Ethernet channel and then one row for each possible device. Each row shows the size and the update time of the channel. In the screenshot there are 3044 bytes sent every ms for the Ethernet channel and 68 bytes every ms for the ibaBM-DDCS device on address 2.

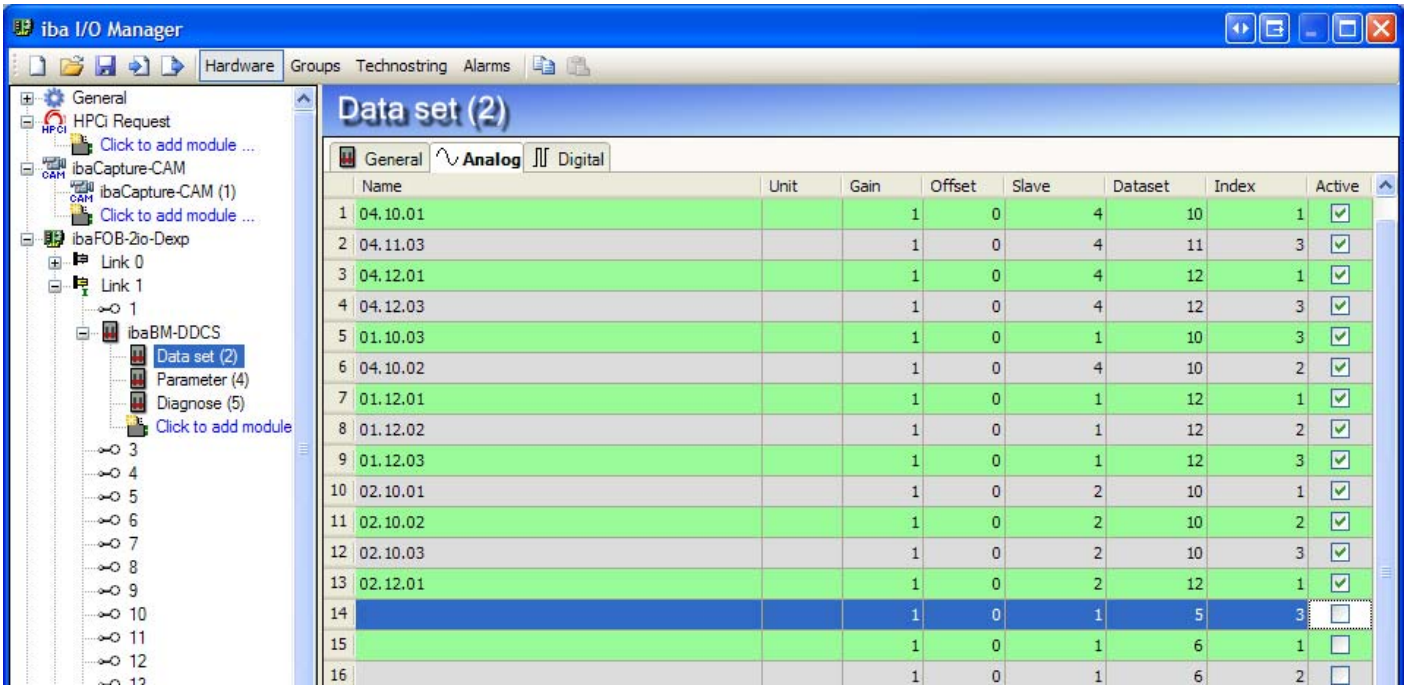
3 ibaBM-DDCS device

The ibaBM-DDCS device is used to measure on the ABB DDCS bus. It is an improvement over the DDCSM device. The ibaBM-DDCS device has 3 possible connections to DDCS busses.



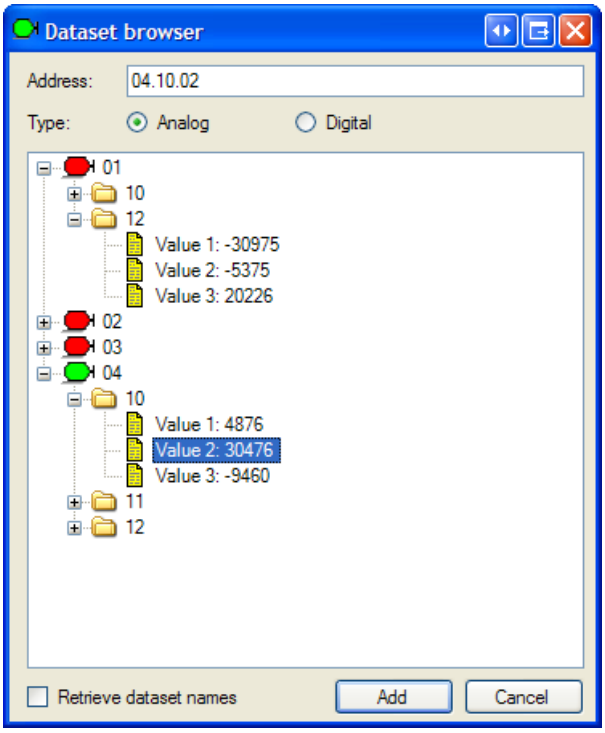
3.1 Datasets

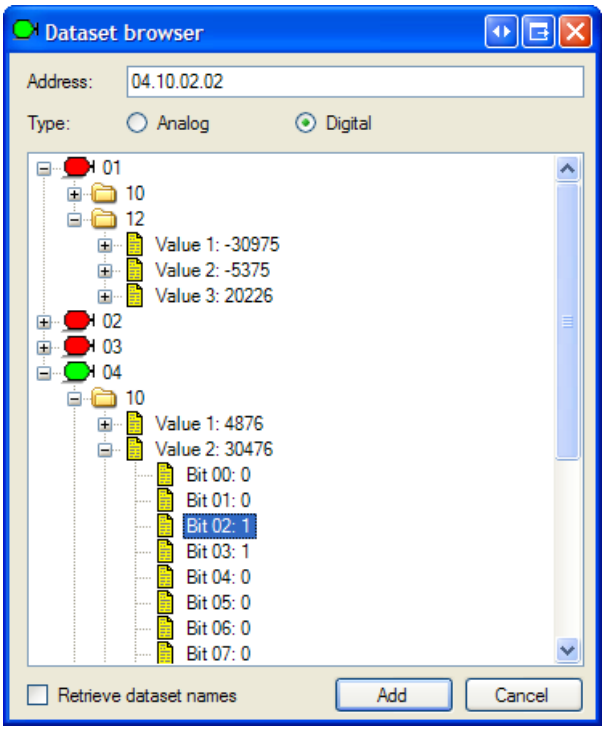
The first connection is used to measure data sets. These are on channel 0 (CH0) according to the ABB terminology. On the general tab of the ibaBM-DDCS device you can enter the baudrate of this channel and the timeout. The datasets are exchanged periodically on the bus. The DDCS device is monitoring this bus and sniffing the data sets. The DDCS doesn't put anything on the DDCS bus for data sets. This was also what the original DDCSM device was measuring. On the original DDCSM the measured data set addresses were fixed. In the current version you can configure the data set address you want to measure. You do this by adding one or more data set modules.



On the general tab you can enter the number of analog and digital signals. For each signal you have to provide the slave number, data set number and index. For the digital signals you also have to provide the bit number. Data set values are 16 bit integers.

There is also a browser for the data sets. You can open it by clicking the Browse hyperlink on the general tab of the data set module. The browser only works if the acquisition is not running. If the acquisition is running the ibaPDA will ask the user if he wants to stop the acquisition.



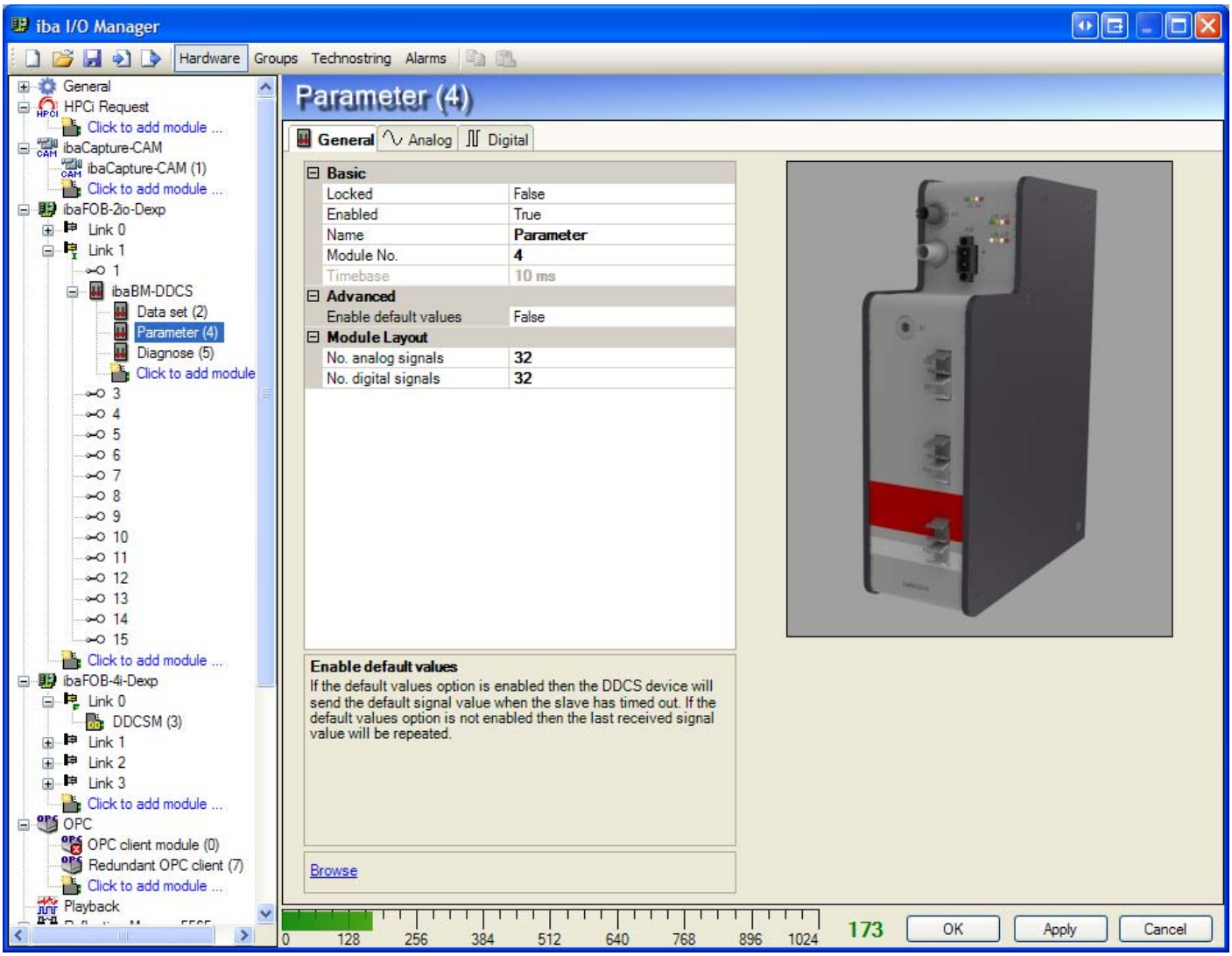


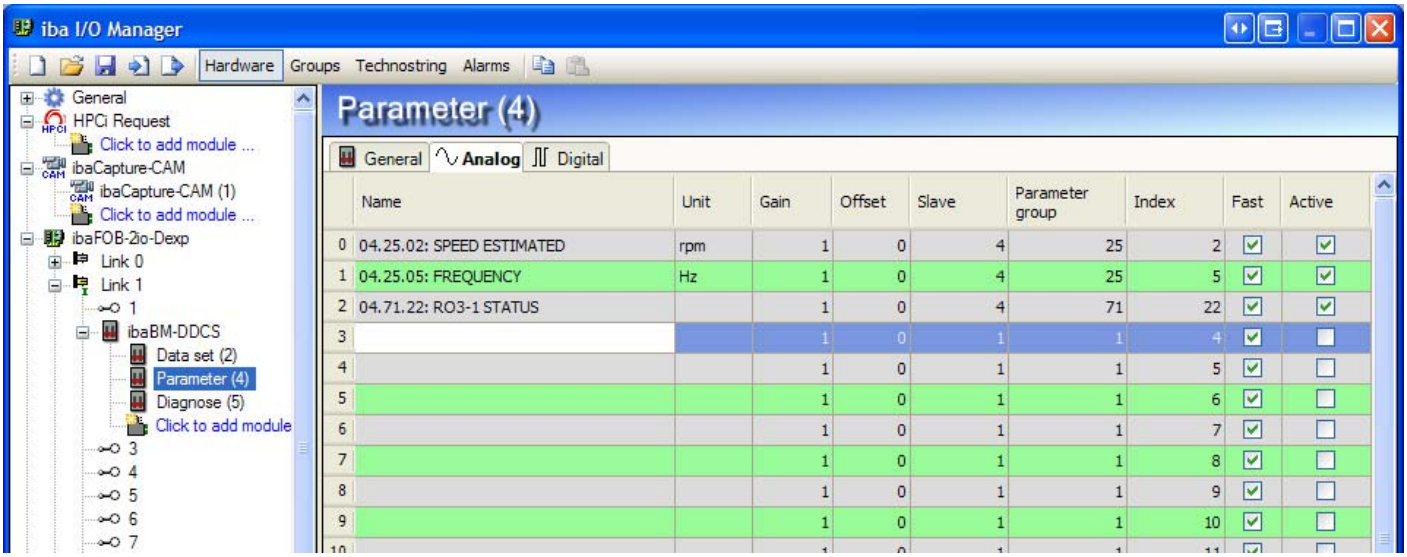
The data set browser shows a tree with all datasets seen on the DDCS bus. The first level contains the slaves. A green icon means that the slave is responding. A red icon means it is not responding. Under a slave you see a number of datasets. Each dataset contains 3 16 bit integer values. The current value is also shown in the browser. If you check the

“Analog” radio button then you can add analog values to the analog signals. If you check the” Digital” radio button then the analog values are expanded into 16 bits. You are then able to add bits to the digital signals.

3.2 Parameters

The second connection is used to measure parameters. These parameters are on channel 3 (CH3) according to the ABB terminology. On the general tab of the ibaBM-DDCS device you can enter the baudrate of this channel and the timeout. The parameters use a polling mechanism. If you want to measure a parameter then the DDCS device will request the value of the parameter periodically. So the DDCS device will be active on the DDCS bus. If you want to measure parameters then you have to add one or more parameter modules.



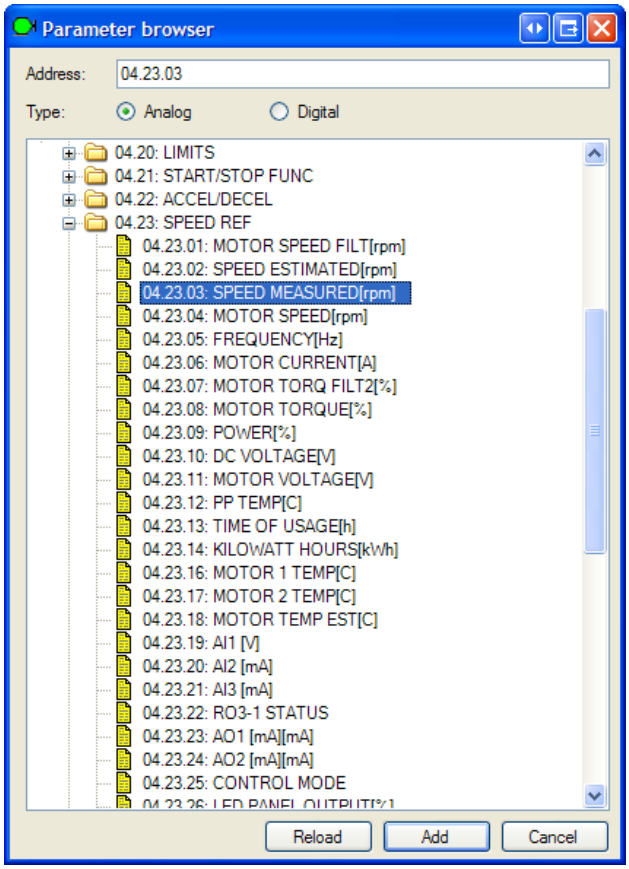


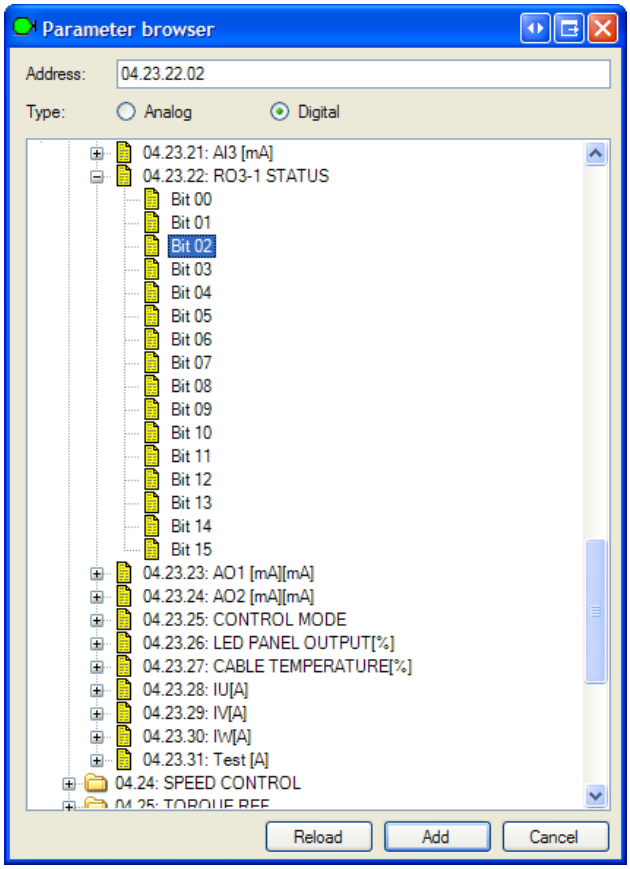
On the general tab you can enter the number of analog and digital signals. For each signal you have to provide the slave number, parameter group number and index. For the digital signals you also have to provide the bit number. Parameter values are 32 bit floating-point values. The fast property determines how fast the parameter is polled. The polling uses the following algorithm:

Loop over all slow signals
{
    Sample all fast signals;
    Sample the next slow signal;
}

This algorithm means that for each slow signal all fast signals are sampled.

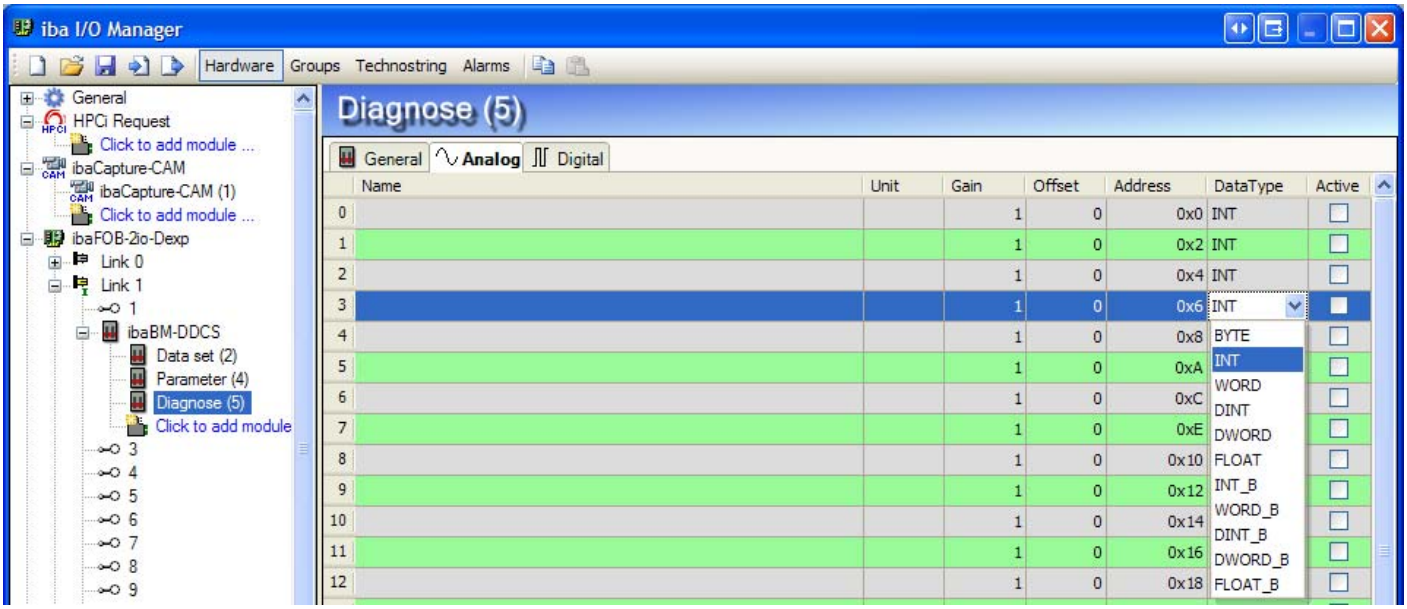
There is also a browser for the parameters. You can open it by clicking the Browse hyperlink on the general tab of the parameter module. The browser only works if the acquisition is not running. If the acquisition is running the ibaPDA will ask the user if he wants to stop the acquisition.





The parameter browser shows a tree with all available parameters. The tree is built up dynamically by requesting the available parameter groups and parameters from the slave when you drill down in the tree. If you check the “Analog” radio button then you can add analog values to the analog signals. If you check the” Digital” radio button then the analog values are expanded into 16 bits. You are then able to add bits to the digital signals.

3.3 Diagnose

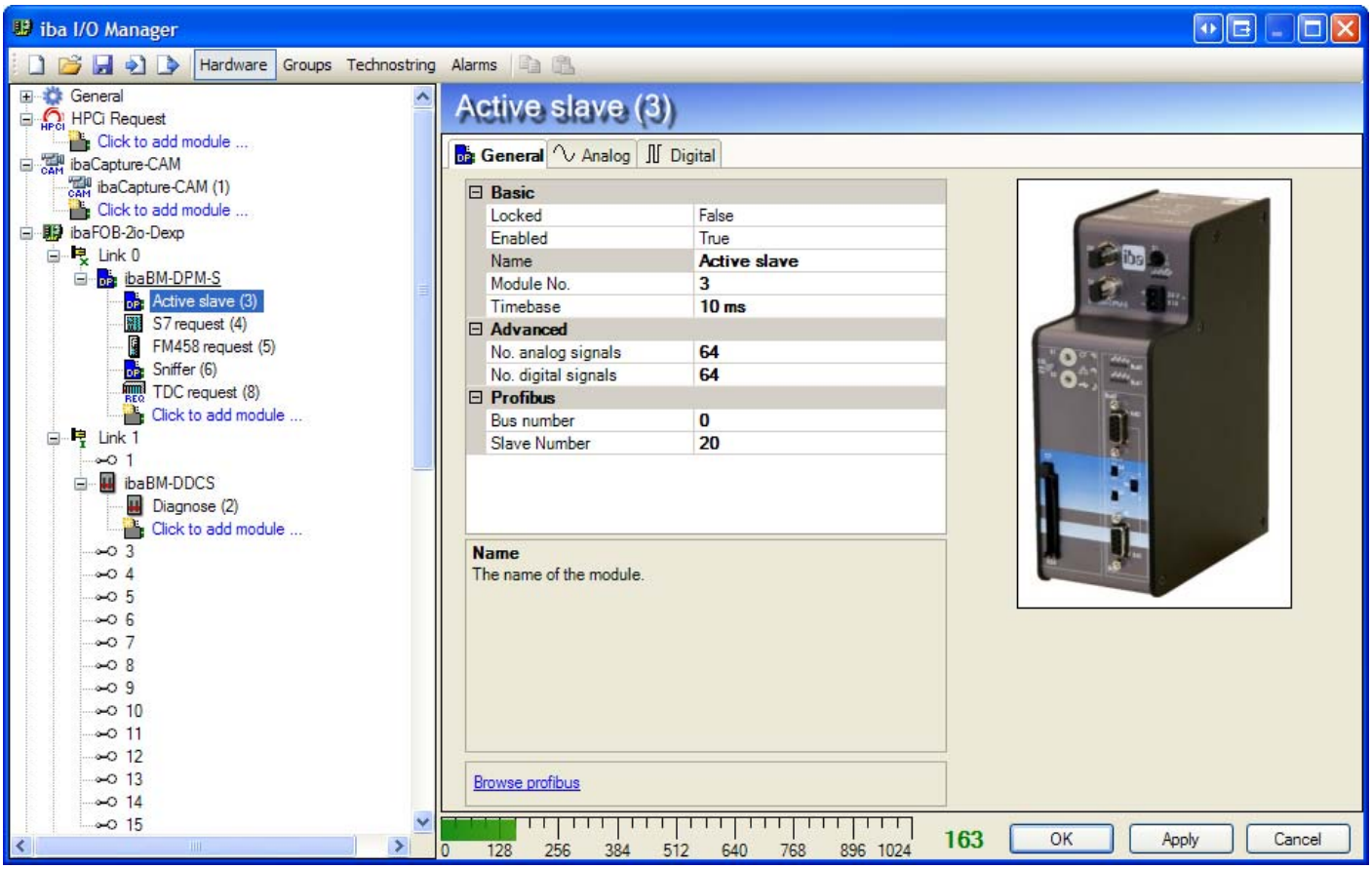


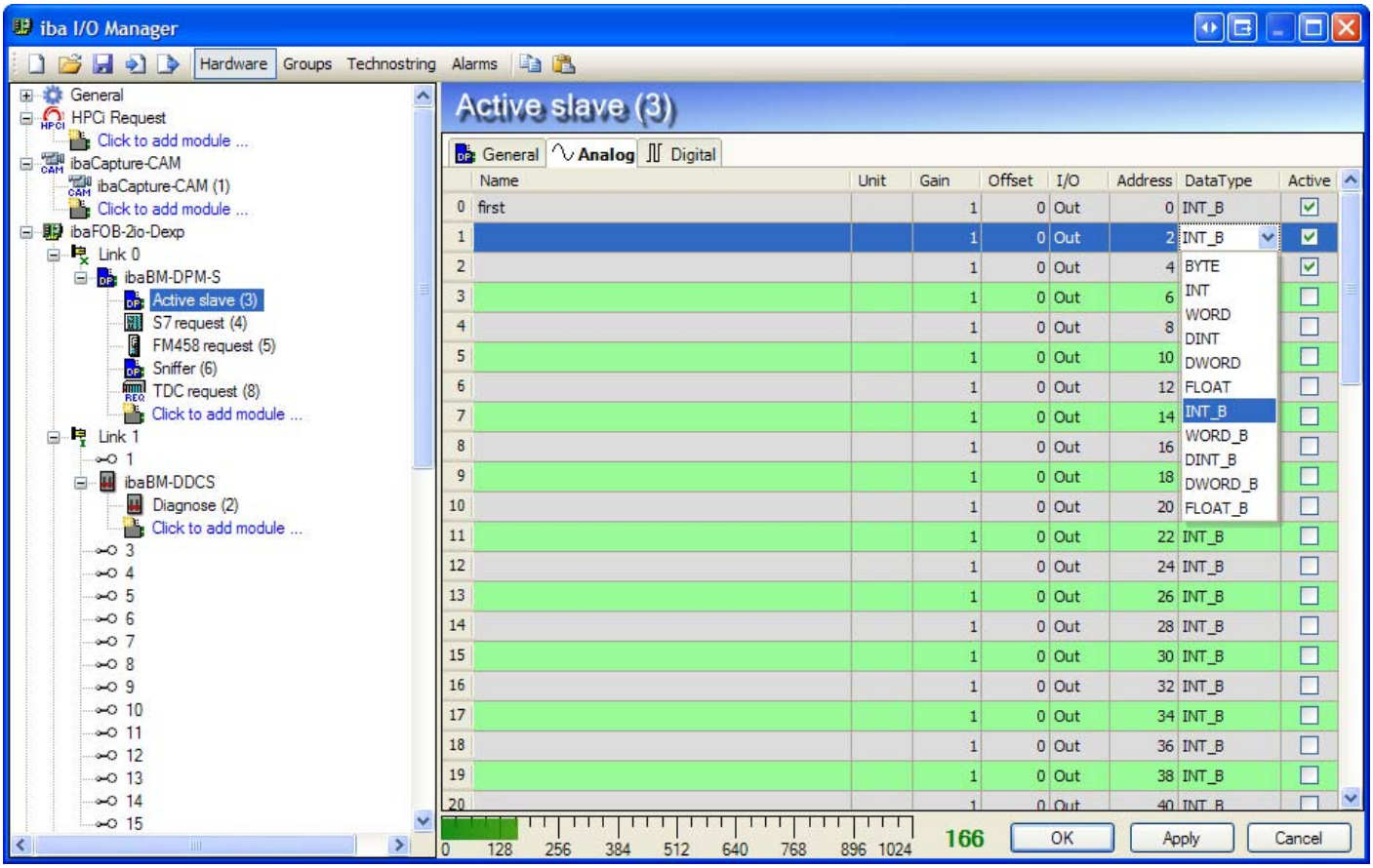
There is also a third module type called diagnose module. This module can be used to measure data from the internal memory of the DDCS device. Currently it is only used for troubleshooting by iba support.

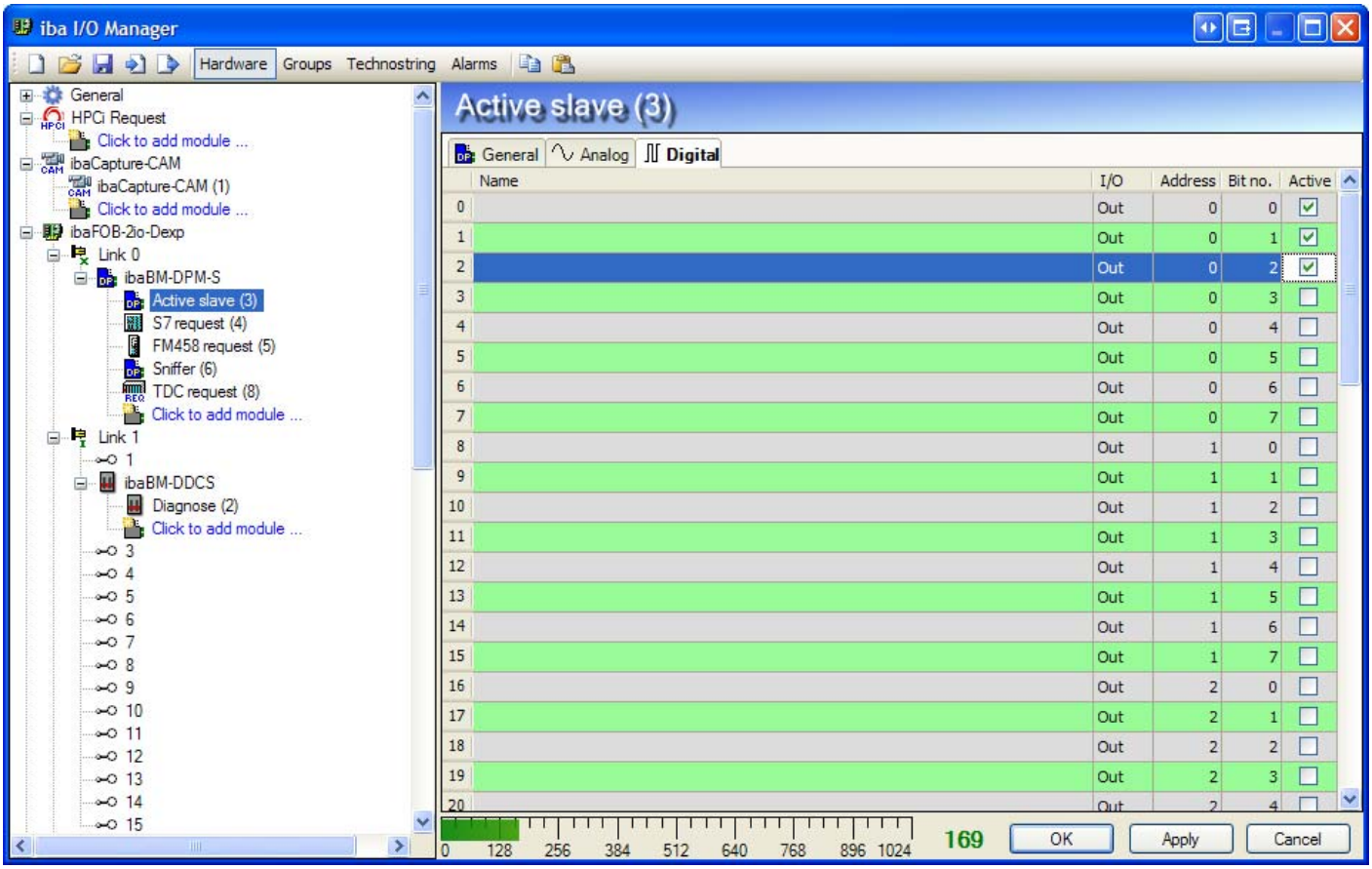
4 ibaBM-DPM-S changes

4.1 Active slave module

There is a new module type: active slave. This module creates an active slave. You specify the bus and the slave number of the active slave on the general tab.



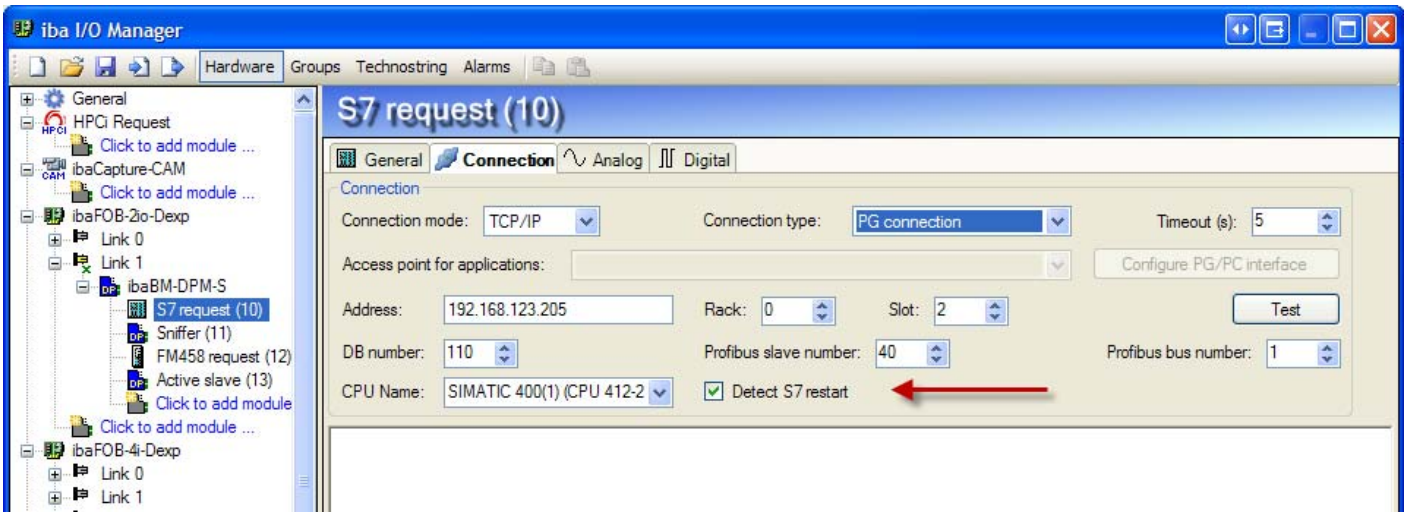

iba I/O Manager
Hardware
Groups
Technostring
Alarms
General
HPCi Request
Click to add module ...
ibaCapture-CAM
ibaCapture-CAM (1)
Click to add module ...
ibaFOB-2io-Dexp
Link 0
ibaBM-DPM-S
Active slave (3)
S7 request (4)
FM458 request (5)
Sniffer (6)
TDC request (8)
Click to add module ...
Link 1
1
ibaBM-DDCS
Diagnose (2)
Click to add module ...
3
4
5
6
7
8
9
10
11
12
13
14
15
Active slave (3)
General
Analog
Digital
Name
Unit
Gain
Offset
I/O
Address
DataType
Active
0 first 1 0 Out 0 INT_B
1 1 0 Out 2 INT_B
2 1 0 Out 4
3 1 0 Out 6
4 1 0 Out 8
5 1 0 Out 10
6 1 0 Out 12
7 1 0 Out 14
8 1 0 Out 16
9 1 0 Out 18
10 1 0 Out 20
11 1 0 Out 22 INT_B
12 1 0 Out 24 INT_B
13 1 0 Out 26 INT_B
14 1 0 Out 28 INT_B
15 1 0 Out 30 INT_B
16 1 0 Out 32 INT_B
17 1 0 Out 34 INT_B
18 1 0 Out 36 INT_B
19 1 0 Out 38 INT_B
20 1 0 Out 40 INT_B
BYTE
INT
WORD
DINT
DWORD
FLOAT
INT_B
WORD_B
DINT_B
DWORD_B
FLOAT_B
0 128 256 384 512 640 768 896 1024
166
OK
Apply
Cancel


iba I/O Manager
Hardware
Groups
Technostring
Alarms
General
HPCi Request
Click to add module ...
ibaCapture-CAM
ibaCapture-CAM (1)
Click to add module ...
ibaFOB-2io-Dexp
Link 0
ibaBM-DPM-S
Active slave (3)
S7 request (4)
FM458 request (5)
Sniffer (6)
TDC request (8)
Click to add module ...
Link 1
1
ibaBM-DDCS
Diagnose (2)
Click to add module ...
3
4
5
6
7
8
9
10
11
12
13
14
15
Active slave (3)
General
Analog
Digital
Name
I/O
Address
Bit no.
Active
0 Out 0 0
1 Out 0 1
2 Out 0 2
3 Out 0 3
4 Out 0 4
5 Out 0 5
6 Out 0 6
7 Out 0 7
8 Out 1 0
9 Out 1 1
10 Out 1 2
11 Out 1 3
12 Out 1 4
13 Out 1 5
14 Out 1 6
15 Out 1 7
16 Out 2 0
17 Out 2 1
18 Out 2 2
19 Out 2 3
20 Out 2 4
0 128 256 384 512 640 768 896 1024
169
OK
Apply
Cancel

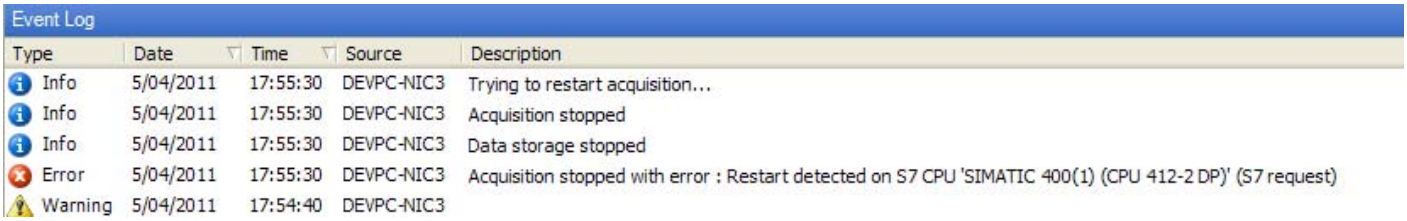
For the analog signals you have to provide the I/O type, address and datatype. Bus number and slave number are taken from the general properties. For the digital signals you have to provide the I/O type, address and bit number.

4.2 S7 request restart detection

S7 request via DPM-S communicates with the S7 and provides a list of operands that an agent on the S7 needs to send to an active slave on the DPM-S. When an S7 CPU is restarted then the list of operands is cleared and the S7 doesn't send the requested operands anymore. In ibaPDA versions before 6.26.0 this was not detected and the user had to restart the acquisition in ibaPDA manually to make the S7 send the requested operands again. Now in 6.26.0 ibaPDA can recognize if the S7 CPU has restarted. When ibaPDA recognizes an S7 CPU restart then the acquisition will be stopped and automatically restarted so that the S7 will send the requested operands again. You can enable this feature on the connection tab of an S7 request module.

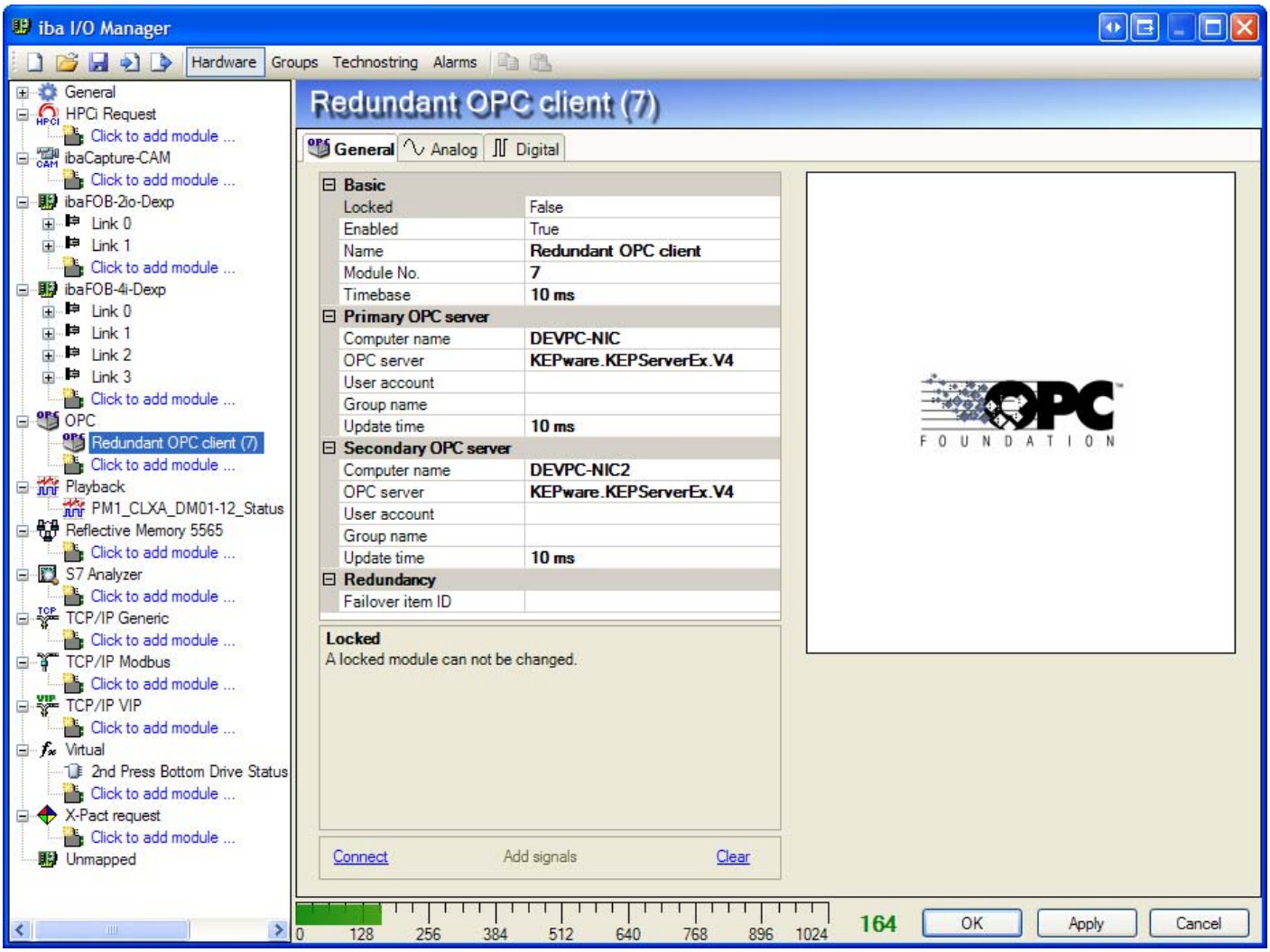


A detected restart is also logged in the event log of the server.

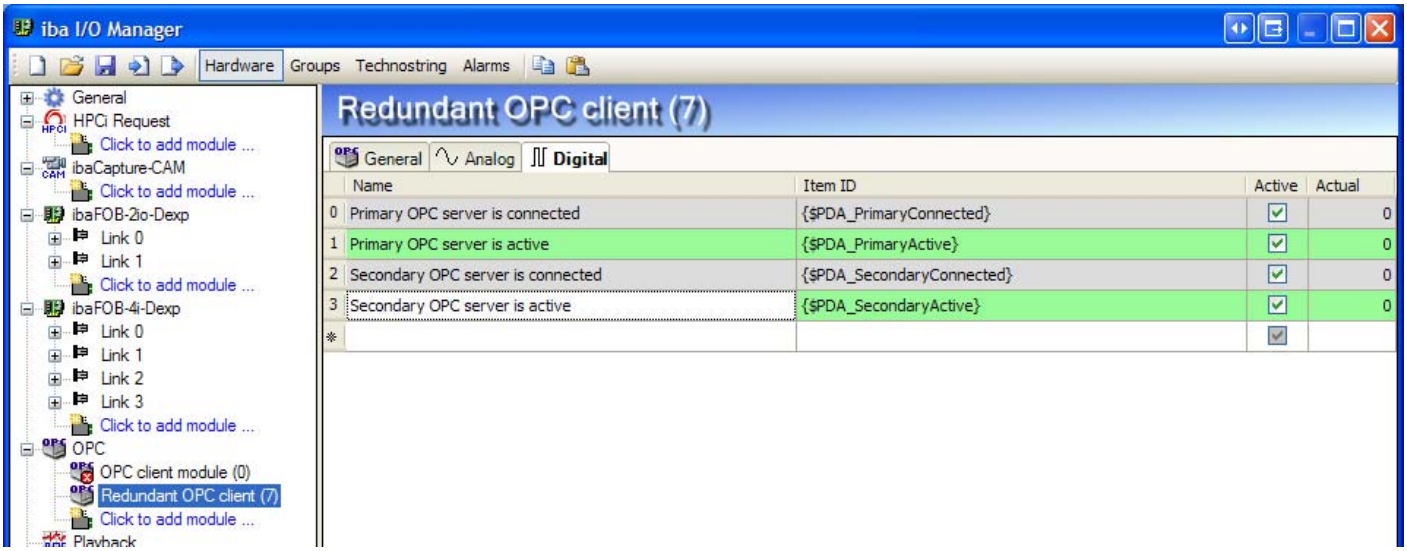


5 Redundant OPC client module

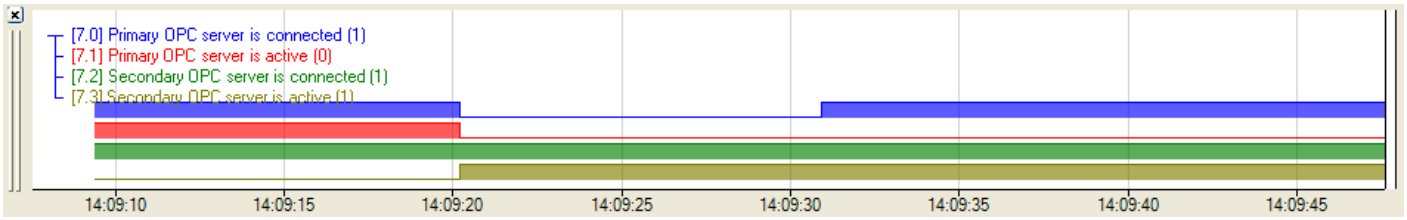
There is a new type of OPC module: redundant OPC client module. The user can configure 2 OPC servers: a primary and secondary. IbaPDA will measure from one of the OPC servers. When the connection to the active OPC server is lost then ibaPDA will automatically switch to the other OPC server.



The “Connect” hyperlink will connect to the primary OPC server. Click the “Add signals” hyperlink to browse the primary OPC server items. The secondary OPC server must have the same items. The browsing and adding of items is completely the same as for a normal OPC client module.



There are 4 predefined digital signals. These signals can't be deleted. They provide information about the operation of the redundant client module. The first signal shows if the primary OPC server is connected. The second signal shows if the primary OPC server is active. An OPC server is active if the OPC server is connected and ibaPDA is using the values coming from that OPC server. The third signal shows if the secondary OPC server is connected. The fourth signal shows if the secondary OPC server is active.



The screenshot shows that both OPC servers are connected and that the primary is active. At 14:09:20 the connection to the primary OPC server is lost and the secondary OPC server is made active. After 10 seconds the primary OPC server is reconnected but the secondary will remain active. An OPC server will remain active as long as the connection is ok.

If you want to control which OPC server is active then you have to use the Failover item ID. The item is constantly monitored on both OPC servers. The OPC server where the value of the failover item is not zero will be active. If the value of the failover item is the same on both servers then the active OPC server will not change. The following screenshot shows a switch from primary to secondary because the failover item changed from 0 to 1 on the secondary OPC server and it changed from 1 to 0 on the primary.