

IQS2 Reference

Software Handbook
Protocol Documentation

Contents

1. Introduction.....	3
2. Installation.....	3
a. System requirements.....	3
b. Installation procedure.....	3
3. Configuration.....	3
4. Graphical user interface.....	4
5. Functional principle of the IQS.....	5
a. Brief description of IQS.....	5
6. Protocol.....	6
a. Basics.....	6
b. 6.2 IQS-Commands.....	6
go.....	6
mstat.....	7
tstat.....	9
confirm.....	9
drop.....	9
help.....	10
quit.....	10

1. Introduction

The IQS server is an optional software component for storage machines made by the company INTERTEX and corresponding machines manufactured under license. It provides an interface which allows the MP-series to be controlled over a network.

As the controls use a RS232 or RS485 interface with a very complex protocol, it was desirable that a simpler interface be made available. For this purpose TCP/IP has been chosen as the transport layer.

2. Installation

a. System requirements

Hardware (minimum requirements)

- 1 GHz Pentium processor
- 256 MB RAM
- 50 MB free hard drive space
- RS232 compatible interface ("COM-Port")
- 10Mbit Ethernet network card

Software

- Windows® NT ,2000, XP
- If the .NET-Framework 2 is not installed yet, you must run the file dotnetfx.exe (approx. 20 MB). This can be downloaded from the following link:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5>

b. Installation procedure

The IQS2 is installed from a CD-ROM as follows:

1. Insert the CD into the drive
2. Open the CD-ROM
3. Run Setup.exe
4. Follow the instructions of the installation assistant

3. Configuration

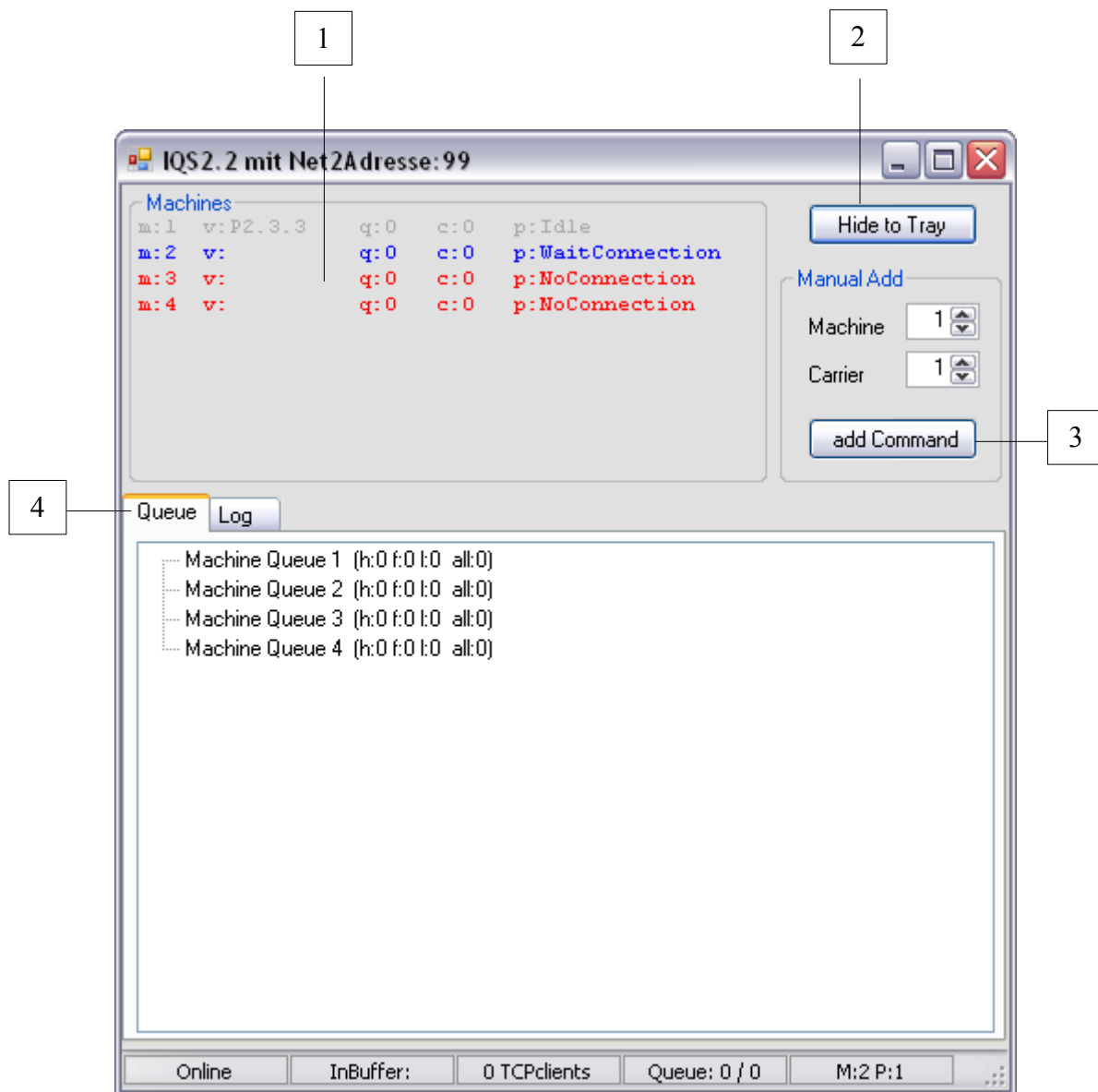
The configuration of the IQS2 is saved in the file „IQS2.4.exe.config“. You get a preconfigured file from INTERTEX, but eventually you have to set the COM-Port correctly. The file can be opened with an editor and it is in the XML format.

For example, if you want to change the COM-Port, take the following changes in the file.

```
<setting name="ComPort" serializeAs="String">  
  <value> Com1 </value>  
</setting>
```

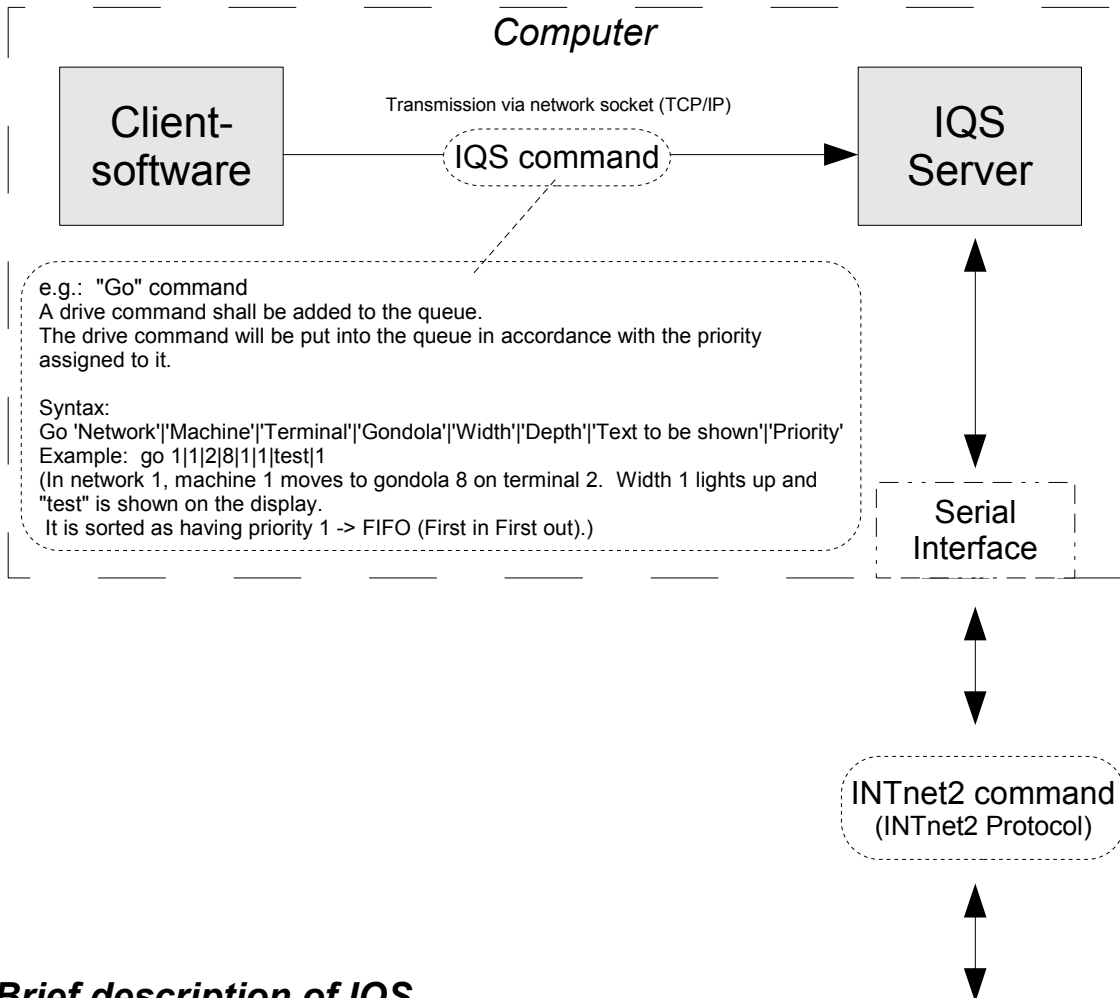
Only the value between <value> and </value> is allowed to be changed!

4. Graphical user interface



1. Shows the status of each machine.
2. With this button the IQS2 can be hidden in the traybar.
3. Here you can add manual instructions to the IQS2.
4. Here you can change between 2 different windows.
 - Queue: shows the upcoming queue of the machines.
 - Log: Here you can view the functions, important for debugging.

5. Functional principle of the IQS



e.g.: "Go" command
 A drive command shall be added to the queue.
 The drive command will be put into the queue in accordance with the priority assigned to it.

Syntax:
 Go 'Network'|'Machine'|'Terminal'|'Gondola'|'Width'|'Depth'|'Text to be shown'|'Priority'
 Example: go 1|1|2|8|1|1|test|1
 (In network 1, machine 1 moves to gondola 8 on terminal 2. Width 1 lights up and "test" is shown on the display.
 It is sorted as having priority 1 -> FIFO (First in First out).)

a. Brief description of IQS

The IQS software is fundamentally a protocol converter with a queue function, which allows the customer to control more easily the machine by means of an INT-MP control. The IQS is configurable and easy to control. It is connected to a configurable port of the PC on which it is installed, and communicates with the INT-MP over the serial interface. Thus you can use your existing network to control the INTERTEX machines.

Its no more necessary to program the complex serial protocol by means of CRC, ACK etc. - the client software sends just one drive command to the queue and the IQS does the rest.

The customer's client software can normally be adapted; in some cases, a detour by means of ASCII-data must be made.

6. Protocol

a. Basics

Each interaction with the IQS server begins with the creation of a network connection over TCP with the corresponding port. After this, normally an IQS command will be sent as a complete string to the IQS. The server then send the corresponding success or error code back.

After this, the corresponding command will be sent from the IQS to the INT-MP. This requires of course that the INT-MP is online.

Individual arguments must be separated by the separation sign "|".
The ASCII value for "|" is 7c (hex) or 124 (dec).

b. 6.2 IQS-Commands

go

Synopsis:

```
go|<net>|<machine>|<terminal>|<carrier>|<width>|<depth>|<usertext>|<priority>
```

Explanation:

This command places a drive command in the queue.

Arguments:

- <net> which network to use is chosen here
- <machine> the target machine which contains the article
- <terminal> the terminal where the goods shall be delivered
- <carrier> the target carrier which contains the article
- <width> 1st axis of the position to be illuminated
- <depth> 2nd axis of the position to be illuminated
- <usertext> text to be shown in the display (2x20 lines)
- <priority> Priority is determined here. There are 3 different priorities:

0 -> Low Priority is put in last position in the queue.

1 -> FIFO tasks are dealt within the order in which they are received

2 -> High Priority is put at the front of the queue, in front of the tasks with priorities 0 and 1 but behind other priority 2 tasks which are already in the queue.

For a "go" command to be correctly carried out, all parameters entered must correspond correctly with the syntax order and with the configuration of the machine(s). Should they not correspond correctly, the IQS will return an error message.

Example 1:

Machine 2 from network 1 is to position load 5 at terminal 1.

5 is given as the width indication and 1 as the depth indication, so that width light 5 and depth light 1 light up. The text is given as "0204025.M4x25.....remove....50.pieces.....", so as to use the 2 lines of 20 characters as a helpful guide. Priority is set to 2, i.e. high.

Correct command:

```
go11121115151110204025.M4x25.....remove..50.pieces...12
```

```
2011OK, added to queue
```

The command has been added to the queue.

Incorrect command:

```
go11121115151110204025.M4x25.....remove..50.pieces...12
4031Argument error in <machine>
```

The same command as above, except that in this case machine 2 does not exist and the command is thus not accepted.

Example 2:

Machine 2 from network 1 is to position load 5 at terminal 1.

0 is given as both width and depth indicator, as this option is not available on the machine in this example. No text is to be shown and so remains empty. Priority is set as 1, i.e. FiFo (First in First out).

Correct command:

```
go111211151010111
2011OK, added to queue
```

The command has been added to the queue.

Incorrect command:

```
go11121115101011
4021Number of arguments error
```

The <usertext> argument has been left out and so the command has not been accepted.

Example3: restore command for the Lift

At Machine 1 from network 1 at terminal 1 the carrier will be brought back to its original position. This command is only for the Lift.

Correct command:

```
go111111101010111
2011OK, added to queue
```

The command has been added to the queue.

mstat

Synopsis:

mstat|<net>|<machine>

Explanation:

The machine status is reported here. The report appears as follows:

The report is divided into 3 lines, with message numbers 210, 211 and 212.

- Message number 210 is always statically reported. It appears as follows:
2101Status of the Machine
- Message number 211 reports the phase information as numerical value and as text. It appears as follows:
2111Phaseinfo1<Phase number>|<Phase text>
Which phase belongs to wich number, can be
It must be noted that the phase between processes between Lift(Li) and Paternoster(Pn) differ.
The following table shows the processes wich are allowed for each machinetype. They are signed with a „X“.

Nummer	Text	Kommentar	Pn	Li
0	NoConnection	(A package is sent to the controller)	x	x
1	WaitConnection	(Waiting for the ACK)	x	x
2	WaitVersion	(Waiting for the version information)	x	x
3	Idle	Normal state (No action)	x	x
11	SendPosCommand	(The package for positioning is sent to the controller)	x	
12	WaitACK	(Waiting for the ACK of the SendPosCommands)	x	
13	WaitOK	(Waiting for the OK)	x	
14	WaitPosReached	Machine is running (Waiting for reaching the position)	x	
15	WaitConfirm	Storage-process is in proceed (After that, confirm the completion)	x	
16	PosDone	Done (The process is complete and jumps back to phase 3(Idle))	x	
21	StatusRequest	Waiting for the status of the machine (carrier taken / not taken)		x
22	StatusWaitACK	(Waiting for the ACK)		x
23	StatusWaitData	(Waiting for the requested status data)		x
31	ResCmdLiftPos	(If the carrier is already taken, the restore command is sent to the controller)		x
32	ResWaitACK	(Waiting for the ACK of the ResCmdLiftPos)		x
33	ResWaitOK	(Waiting for the OK)		x
34	ResWaitPosReach	Machine is running (Waiting for the restore of the carrier)		x
35	ResWaitConfirm	Storage-process is in proceed (Completion is confirmed automatically by the controller)		x
41	GetCmdLiftPos	(Sending of the package of positioning to the controller)		x
42	GetWaitACK	(Waiting for the ACK of the GetCmdLiftPos)		x
43	GetWaitOK	(Waiting for the OK)		x
44	GetWaitPosReach	Machine is running (Waiting, that the carrier is at the right output station)		x
45	GetWaitConfirm	Storage-process is in proceed (After that, confirm the completion)		x
46	Done	Done (The process is complete and jumps back to phase 3(Idle))		x

- Message number 212 reports the number of entries in the queue. It appears as follows:

```
212|Entrys in Queuel <number of entries in queue>
```

- The Message number 213 reports whether the latest stock transaction was confirmed (Enter) or canceled (CE). It appears as follows:

```
213|Last Transaction Confirmation
```


Example:

```
mstatl 1|1
210|Status of the Machine
211|Phaseinfo| 0|NoConnection
212|Entry in Queue| 0
213| Last Transaction Confirmed|OK
```

In this example there is no connection to the INT-MP and there are no entries in the queue.

tstat**Synopsis:**

```
tstat|<net>|<machine>|<terminal>
```

Explanation:

It returns the current carrier position of the Terminal. It appears as follows:

Lift:

```
tstatl111111
215|Terminal 111001
In this example carrier 1001 is located at terminal 1.
```

```
tstatl111111
215|Terminal 110
In this example there is no carrier at terminal 1.
```

Paternoster:

```
tstatl111111
215|Terminal 111
In this example the paternoster is at postion 1.
```

confirm**Synopsis:**

```
confirm|<net>|<machine>
```

Explanation:

A machine which is in phase 8 (WaitConfirm – waiting for a confirmation), must be confirmed from the PC. Thus the IQS2 initiate the confirmation at the machine and the next record of the queue can be handled. The return always looks like this:

```
231|Remote- Confirm will be done
```

drop**Synopsis:**

```
drop|<net>|<machine>
```

Explanation:

The "drop" command clears the machine's current queue. If the machine is already arriving at a carrier, this command will be completed.

help

Synopsis:
help

Explanation:

This command displays the currently available commands and their respective syntax. This command can only be used when a manual connection has been established to the IQS through a terminal program, and one wants to see an overview of the commands and their synopses.

quit

Synopsis:
quit

Explanation:

This command breaks off the connection to the server.

Example:

```
quit  
2211Closing Connection
```