

IQS2 Referenz

Software Handbuch
Dokumentation des Protokolls

Inhaltsverzeichnis

1. Einleitung.....	3
2. Installation.....	3
a. Systemvoraussetzungen.....	3
b. Installationsablauf.....	3
3. Konfiguration.....	3
4. Grafische Benutzeroberfläche.....	4
5. Funktionsprinzip der IQS.....	5
a. Kurzbeschreibung der IQS.....	5
6. Protokoll.....	6
a. Grundsätzliches.....	6
b. IQS-Befehle.....	6
go.....	6
mstat.....	7
tstat.....	9
confirm.....	9
drop.....	9
help.....	10
quit.....	10

1. Einleitung

Der IQS Server ist eine optionale Software Komponente für Lagermaschinen der Firma INTERTEX und entsprechende in Lizenz gefertigte Maschinen. Sie stellt eine Schnittstelle dar, um die Maschinensteuerungen der „MP“-Serie netzwerkfähig zu machen.

Da die Steuerungen eine RS232 oder RS485 Schnittstelle mit einem sehr komplexen Protokoll verwenden, kam der Wunsch auf, eine simple Schnittstelle zur Verfügung zu haben. Gewählt wurde hierfür TCP/IP als Transport-Layer.

2. Installation

a. Systemvoraussetzungen

Hardware (Minimalvoraussetzung)

- 1 GHz Pentium Processor
- 256 MB RAM
- 50 MB freier Festplattenspeicher
- RS232 fähige Schnittstelle („COM-Port“)
- 10Mbit Ethernet Netzwerkkarte

Software

- Windows® NT ,2000, XP
- Falls das .NET-Framework 2.0 noch nicht installiert ist, müssen Sie die Datei dotnetfx.exe ausführen. Diese können Sie unter folgendem Link runterladen:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5>

b. Installationsablauf

So installieren Sie den IQS2 von einer CD ROM aus:

1. Legen Sie die CD in Ihr CDRom-Laufwerk ein.
2. Öffnen Sie die CDRom.
3. Führen Sie die Setup.exe aus.
4. Folgen Sie den Anweisungen des Installations-Assistenten.

3. Konfiguration

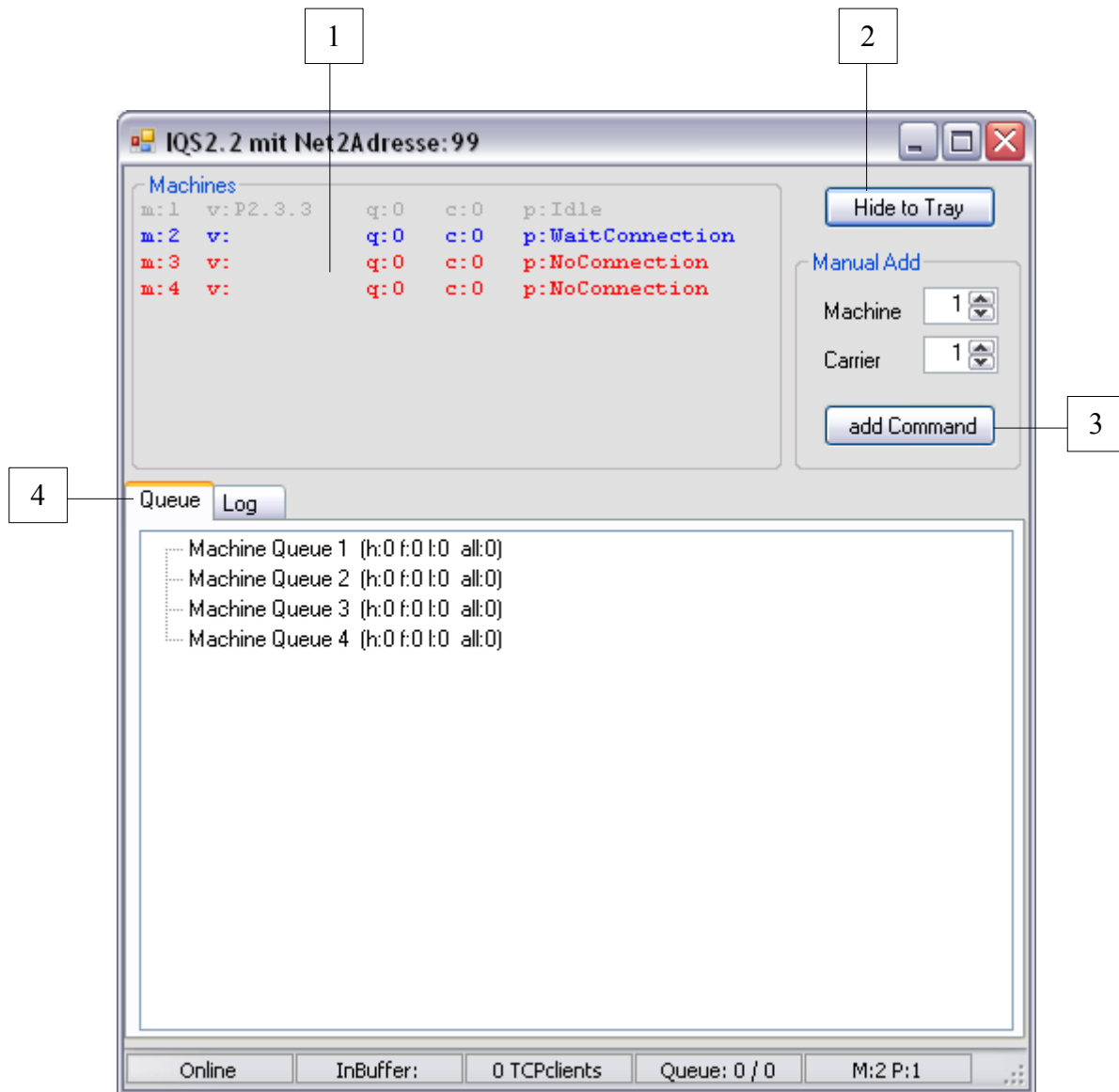
Die Konfiguration der IQS2 ist in der Datei „IQS2.4.exe.config“ gespeichert. Sie erhalten von INTERTEX schon eine vorkonfigurierte Datei, aber eventuell muss z.B. der COM-Port noch korrekt eingestellt werden. Die Datei kann einfach mit einem Editor geöffnet werden und ist im XML-Format gehalten.

Wenn Sie die zum Beispiel den COM-Port ändern wollen so machen Sie die Änderung im folgenden Abschnitt.

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

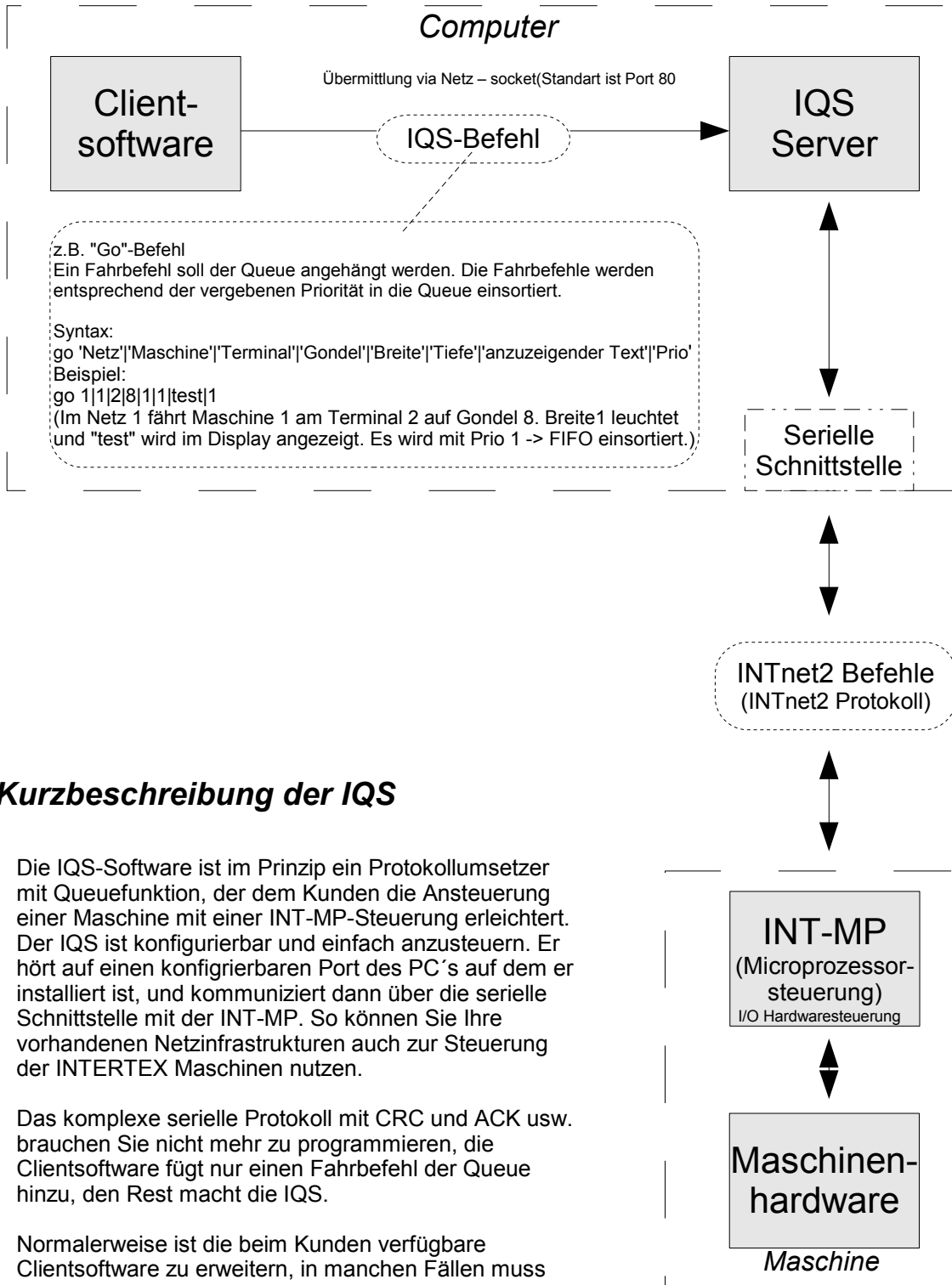
Geändert werden darf nur der Wert zwischen <value> und </value>!

4. Grafische Benutzeroberfläche



- 1.Hier wird die Maschinenstatus der einzelnen Maschinen angezeigt.
- 2.Mit diesem Button kann der IQS2 in der Traybar versteckt werden.
- 3.Hier kann man dem IQS2 manuell Befehle hinzufügen.
- 4.Hier kann man zwischen 2 verschiedenen Fenstern wechseln.
 - Queue: zeigt die anstehenden Queue's der Maschinen an.
 - Log: Hier kann man die Funktion einsehen, wichtig für die Fehlersuche.

5. Funktionsprinzip der IQS



a. Kurzbeschreibung der IQS

Die IQS-Software ist im Prinzip ein Protokollumsetzer mit Queuefunktion, der dem Kunden die Ansteuerung einer Maschine mit einer INT-MP-Steuerung erleichtert. Der IQS ist konfigurierbar und einfach anzusteuern. Er hört auf einen konfigurierbaren Port des PC's auf dem er installiert ist, und kommuniziert dann über die serielle Schnittstelle mit der INT-MP. So können Sie Ihre vorhandenen Netzinfrastrukturen auch zur Steuerung der INTERTEX Maschinen nutzen.

Das komplexe serielle Protokoll mit CRC und ACK usw. brauchen Sie nicht mehr zu programmieren, die Clientsoftware fügt nur einen Fahrbefehl der Queue hinzu, den Rest macht die IQS.

Normalerweise ist die beim Kunden verfügbare Clientsoftware zu erweitern, in manchen Fällen muss ein Umweg über ASCII-Dateien gemacht werden.

6. Protokoll

a. Grundsätzliches

Grundsätzlich beginnt jede Interaktion mit dem IQS Server mit einem Verbindungsaufbau über TCP auf den entsprechenden Port. Danach wird ein IQS-Befehl im Normalfall als kompletter String an den IQS gesendet. Der Server sendet dann den entsprechenden Erfolgs- bzw. Fehlercode zurück.

Danach wird der entsprechende Befehl von dem IQS an die INT-MP gesendet.

Grundsätzlich werden die einzelnen Argumente durch das Trennzeichen „|“ getrennt. Der ASCII-Wert für „|“ ist 7C(hex) bzw. 124(dez).

b. IQS-Befehle

go

Synopsis:

```
go|<net>|<maschine>|<terminal>|<carrier>|<width>|<depth>|<usertext>|<prio>
```

Erklärung:

Mit diesem Befehl wird ein Fahrbefehl in die Queue plaziert.

Argumente:

- <net> hier wählt man aus, welches Netz zum Einsatz kommt.
- <maschine> hier wird die Maschine bestimmt.
- <terminal> Terminal an dem die Ware entnommen werden soll.
- <carrier> Die anzufahrende Position.
- <width> 1. Achse der anzuleuchtenden Position.
- <depth> 2. Achse der anzuleuchtenden Position.
- <usertext> Text, der im Display angezeigt werden soll. (2x20 Zeichen)
- <prio> Hier wird die Priorität festgelegt. Es gibt 3 verschiedene Prioritäten:
 0 -> Low Priorität wird ganz hinten an die Queue angehängt.
 1 -> FIFO die Aufträge werden so abgearbeitet wie sie reinkommen
 2 -> High Priorität wird oben an die Queue angehängt, vor die Aufträge mit der Prio 0 und 1. Allerdings hinter die schon vorhandenen Aufträge mit der Prio 2.

Wenn der „go“-Befehl richtig ausgeführt werden soll so müssen alle übergebenen Parameter mit der Reihenfolge der Syntax und der Konfiguration der Maschine übereinstimmen. Sollte dies nicht der Fall sein, so gibt der IQS eine Fehlermeldung zurück.

Beispiel1:

Aus dem Netz 1 soll die Maschine 2 an dem Terminal 1 den Tragsatz 5 positionieren. Als Breitenanzeige wird die 5 und als Tiefenanzeige die 1 übergeben, damit die Breitenlampe 5 leuchtet und die Tiefenlampe 1. Als Text wird „0204025.M4x25.....entn....50.Stk.....“ übergeben, um die 2 Zeilen a 20 Zeichen als Hilfestellung zu nutzen. Als Priorität wird 2, also High, benutzt.

Korrekter Befehl:

```
go|1|2|1|1|5|1|5|1|1|0|2|0|4|0|2|5|.M4x25|.....entn|....50.Stk|.....|2
```

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

Der Befehl wurde in der Queue plaziert.

Fehlerhafter Befehl:

```
go11121115151110204025.M4x25.....entn....50.Stk.....12
4031Argument error in <machine>
```

Der gleiche Befehl wie oben, nur existiert in diesem Fall die Maschine 2 nicht, der Befehl wurde nicht aufgenommen.

Beispiel2:

Aus dem Netz 1 soll die Maschine 2 an dem Terminal 1 den Tragsatz 5 positionieren.

Als Breiten- und Tiefenanzeige wird jeweils eine 0 übergeben, da diese Option an der Maschine in diesem Beispiel nicht verfügbar ist. Text muss keiner angezeigt werden, bleibt also leer. Als Priorität wird 1, also FiFo, benutzt.

Korrekter Befehl:

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

Der Befehl wurde in der Queue plaziert.

Fehlerhafter Befehl:

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

Es wurde das Argument <usertext> vergessen, der Befehl wurde nicht aufgenommen.

Beispiel3: reiner Restorebefehl für den Lift

Am Netz 1 soll an Maschine 1 / Terminal 1 der Tragsatz an seinen ursprünglichen Ort zurückgebracht werden. Dieser Befehl ist spezifisch für den Lift.

Korrekter Befehl:

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

Der Befehl wurde in der Queue plaziert.

mstat

Synopsis:

mstat|<net>|<machine>

Erklärung:

Es wird hier der Status der Maschine zurückgegeben. Die Rückgabe sieht folgendermaßen aus:

Die Rückgabe teilt sich in 3 Zeilen auf, mit den Meldungsnummern 210, 211 und 212.

- Die Meldung Nummer 210 wird immer statisch zurückgegeben. Sie sieht folgendermaßen aus:
2101Status of the Machine
- Die Meldung Nummer 211 gibt die Phaseninformation zurück, als Nummernwert und auch als Text. Sie sieht folgendermaßen aus:
- 2111Phaseinfo|<Phasennummer>|<Phasentext>
Welche Phase welche Nummer hat, kann aus der nachfolgenden Tabelle entnommen werden. Dabei muss beachtet werden, dass die Phasenabläufe zwischen Lift(Li) und Paternoster(Pn) sich unterscheiden. In der nachfolgenden Tabelle wurden die Phasen, die für den Maschinentyp möglich sind, mit einem „x“ gekennzeichnet:

Nummer	Text	Kommentar	Pn	Li
0	NoConnection	(Es wird ein Paket an die Steuerung geschickt)	x	x
1	WaitConnection	(Es wird auf das ACK gewartet)	x	x
2	WaitVersion	(Es wird auf die Versioninfo gewartet)	x	x
3	Idle	Normalzustand (Keine Aktion)	x	x
11	SendPosCommand	(Es wird das Paket zum Positionieren an die Steuerung gesendet)	x	
12	WaitACK	(Es wird auf das ACK des SendPosCommands gewartet)	x	
13	WaitOK	(Es wird auf das OK gewartet)	x	
14	WaitPosReached	Maschine läuft (Es wird auf das Erreichen der Position gewartet)	x	
15	WaitConfirm	Lagervorgang läuft (Danach wird die Fertigstellung bestätigt)	x	
16	PosDone	Fertig (Der Vorgang ist komplett und springt zurück zur Phase 3(Idle))	x	
21	StatusRequest	Es wird der Status der Maschine angefordert (Träger belegt/nicht belegt)		x
22	StatusWaitACK	(Es wird auf das ACK gewartet)		x
23	StatusWaitData	(Warten auf die angeforderten Statusdaten)		x
31	ResCmdLiftPos	(Wenn der Träger bereits belegt ist, wird der Restorebefehl an die Steuerung gesendet)		x
32	ResWaitACK	(Es wird auf das ACK des ResCmdLiftPos gewartet)		x
33	ResWaitOK	(Es wird auf das OK gewartet)		x
34	ResWaitPosReach	Maschine läuft (Warten, daß Träger zurückgelagert wird)		x
35	ResWaitConfirm	Lagervorgang läuft (Fertigstellung wird von der Steuerung selbstständig bestätigt)		x
41	GetCmdLiftPos	(Es wird das Paket zum Positionieren an die Steuerung gesendet)		x
42	GetWaitACK	(Es wird auf das ACK des GetCmdLiftPos gewartet)		x
43	GetWaitOK	(Es wird auf das OK gewartet)		x
44	GetWaitPosReach	Maschine läuft (Warten, daß Träger an die gewünschte Ausgabestation gelagert wird.)		x
45	GetWaitConfirm	Lagervorgang läuft (Danach wird die Fertigstellung bestätigt)		x
46	Done	Fertig (Der Vorgang ist komplett und springt zurück zur Phase 3(Idle))		x

- Die Meldung Nummer 212 gibt die Anzahl der Einträge in der Queue zurück. Sie sieht folgendermaßen aus:

```
212|Entrys in Queue| <AnzahlEinträgeInQueue>
```

- Die Meldung Nummer 213 gibt zurück ob der letzte Lagervorgang mit bestätigt(Enter) oder abgebrochen(CE) wurde. Sie sieht folgendermaßen aus:

```
213|Last Transaction Confirmation|
```


Beispiel:

```
mstatlllll
210lStatus of the Machine
211lPhaseinfo10lNoConnection
212lEntrys in Queue10
213lLast Transaction ConfirmationlOK
```

In diesem Beispiel ist die Verbindung zur INT-MP ist nicht hergestellt und es ist kein Eintrag in der Queue vorhanden.

tstat**Synopsis:**

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

Erklärung:

Es wird die aktuelle Tragsatzposition des Terminals zurückgegeben. Die Rückgabe sieht folgendermaßen aus:

Lift:

```
tstatlllllll
215lTerminal 111001
```

In diesem Beispiel befindet sich der Träger 1001 an Terminal 1.

```
tstatlllllll
215lTerminal 110
```

In diesem Beispiel befindet sich kein Träger an Terminal 1.

Paternoster:

```
tstatlllllll
215lTerminal 111
```

In diesem Beispiel befindet sich der Paternoster aktuell an Position 1.

confirm**Synopsis:**

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

Erklärung:

Eine Maschine, die sich in der Phase 8 (WaitConfirm - auf eine Bestätigung warten) befindet, soll vom PC aus bestätigt werden. Somit veranlasst die IQS2 die Bestätigung an der Maschine und es kann der nächste Queueeintrag bearbeitet werden. Die Rückgabe sieht immer folgendermaßen aus:

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

drop**Synopsis:**

```
drop|<net>|<maschine>
```

Erklärung:

Der „drop“-Befehl löscht die aktuelle Queue der Maschine. Falls die Maschine gerade eine Gondel anfährt, wird dieser Befehl noch vollständig ausgeführt.

help

Synopsis:
help

Erklärung:

Mit diesem Befehl werden die derzeit verfügbaren Befehle mit ihrer jeweiligen Syntax angezeigt. Dieser Befehl wird nur dann gebraucht, wenn über ein Terminalprogramm manuell zum IQS eine Verbindung besteht, und man sich einen Überblick über die Befehle und deren Synopsis verschaffen möchte.

quit

Synopsis:
quit

Erklärung:

Mit diesem Befehl wird die Verbindung zum Server unterbrochen.

Beispiel:

```
quit  
2211Closing Connection
```