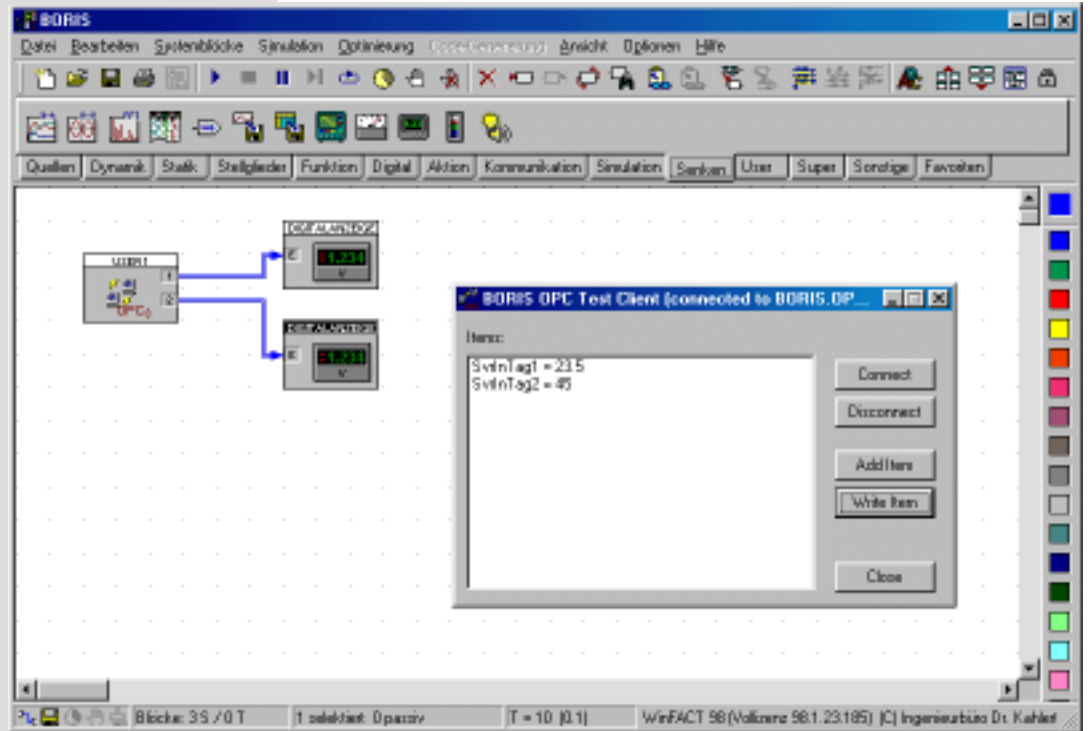


WinFACT 98



Ingenieurbüro Dr. Kahlert

Ludwig-Erhard-Str. 45
D-59065 Hamm

WinFACT 98 OPC-Client/Server-Toolbox

0. Einführung

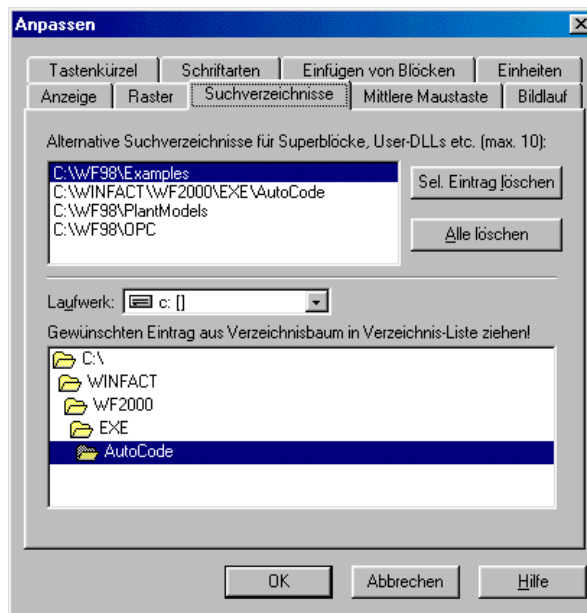
Die *WinFACT OPC-Client/Server-Toolbox* enthält sämtliche Komponenten, um das Blockorientierte Simulationssystem BORIS wahlweise als OPC-Server oder OPC-Client zu betreiben. Dazu zählen:

- Server-Input-Block (BORIS-User-DLL *OPCSERVERIN.DLL*)
- Server-Output-Block (BORIS-User-DLL *OPCSERVEROUT.DLL*)
- Client-Input-Block (BORIS-User-DLL *OPCLIENTIN.DLL*)
- Client-Output-Block (BORIS-User-DLL *OPCLIENTOUT.DLL*)
- BORIS-OPC-Testserver zum Testen von Client-Blöcken (Datei *BORISOPCTESTSERVER.EXE*)
- BORIS-OPC-Testclient zum Testen von Server-Blöcken (Datei *BORISOPCTESTCLIENT.EXE*)

Alle Komponenten befinden sich nach der Installation der Toolbox im Unterverzeichnis *\OPC* des WinFACT-Programmverzeichnisses.

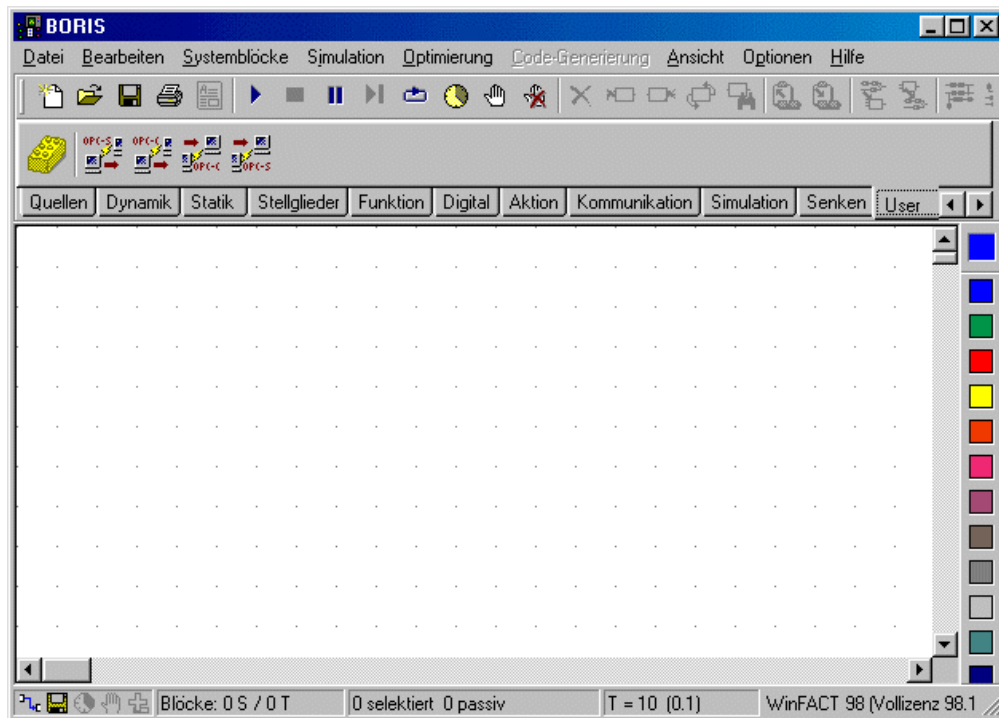
1. Konfigurierung von BORIS

Nach der Installation der Toolbox sollten Sie zunächst das Installationsverzeichnis der Toolbox (standardmäßig *c:\wf98\opc*) unter BORIS als Suchverzeichnis einrichten, damit BORIS die OPC-Blöcke in seiner Systemblock-Toolbar anzeigt. Wählen Sie dazu in BORIS die Menüoption *OPTIONEN | ANPASSEN...* und geben Sie das Verzeichnis im entsprechenden Dialog an.



Einrichten des OPC-Verzeichnisses als Suchverzeichnis unter BORIS

Beim nächsten Starten von BORIS erscheinen die OPC-Blöcke dann wie gewohnt auf der User-DLL-Palette der Systemblock-Toolbar.

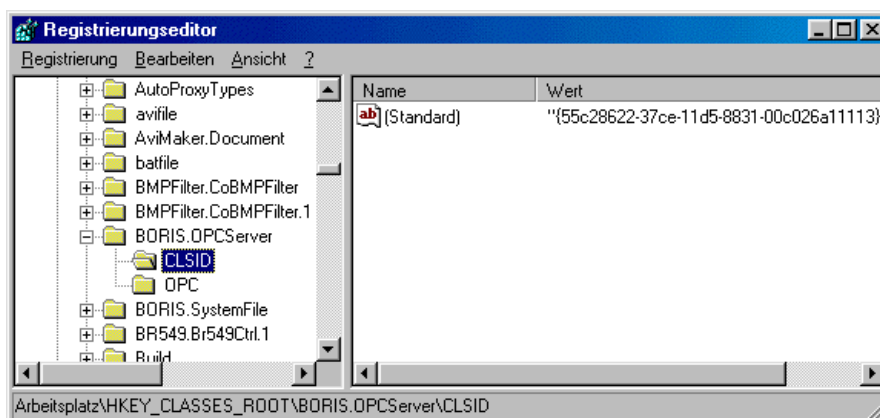


OPC-Blöcke auf der User-DLL-Palette der BORIS-Systemblock-Toolbar

2. Betrieb von BORIS als OPC-Server

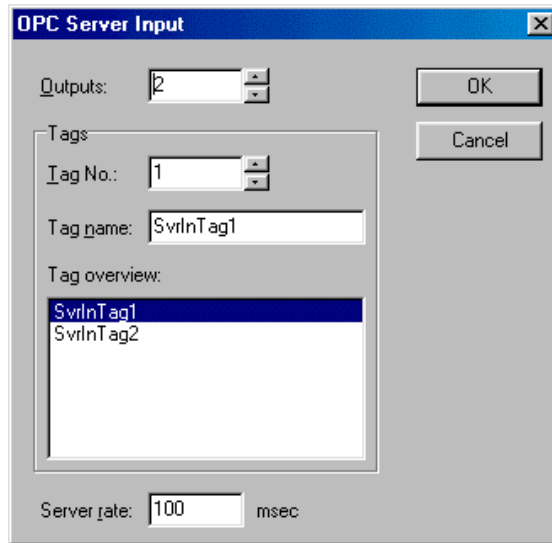
Mit Hilfe des *OPCServerIn*- bzw. *OPCServerOut*-Blocktyps lässt sich BORIS als OPC-Server betreiben. Dies bedeutet, dass Signale aus BORIS heraus mit einem beliebigen OPC-Client gelesen oder aber auch Daten aus einem OPC-Client in eine BORIS-Struktur eingespeist werden können. Selbstverständlich sind beide Blocktypen mehrmals und auch gleichzeitig innerhalb ein- und derselben BORIS-Struktur verwendbar.

Der BORIS-OPC-Server trägt den OPC-Bezeichner *BORIS.OPCServer* und wird beim erstmaligen Einbinden eines OPCServer-Blocks in eine BORIS-Struktur automatisch in der Windows-Registry unter *HKEY_CLASSES_ROOT* registriert.



Der BORIS-OPC-Server-Eintrag in der Windows-Registry

Der *OPCServerIn*-Block liest bis zu 50 Signale über einen OPC-Client ein. Nachfolgende Bildschirmgrafik zeigt den zugehörigen Parameterdialog.

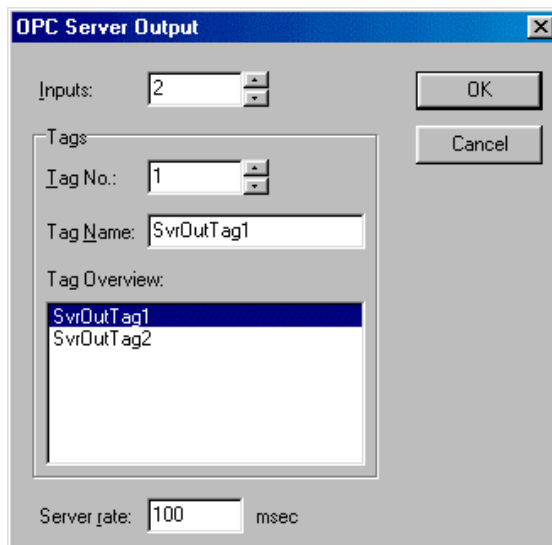


Parameterdialog des OPCServerIn-Blocks

Im *Outputs*-Eingabefeld kann zunächst die Anzahl der Blockausgänge (d. h. der einzulesenden Signale) vorgegeben werden. Die Gruppenbox *Tags* ermöglicht dann die Festlegung der entsprechenden *Tagnames* für die einzelnen Eingänge. Voreingestellt werden die Servertags mit *SvrInTag1*, ... bezeichnet. Im Eingabefeld *Server rate* wird die Serverrate in Millisekunden angegeben.

Zu Beginn der Simulation werden alle Ausgänge des Blocks mit dem Wert -99999 vorbelegt. Dieser Wert wird dann solange beibehalten, bis von einem OPC-Client für den entsprechenden Ausgang ein gültiger Wert geschrieben wird. BORIS interpretiert dabei die Fließkomma-, Ganzzahl- und Booleschen Datentypen als Fließkommazahlen (der boolesche Wert TRUE wird als 1.0, der boolesche Wert FALSE als 0.0 übernommen); ein STRING-Datentyp wird als ungültig und demzufolge mit dem Wert -99999 interpretiert.

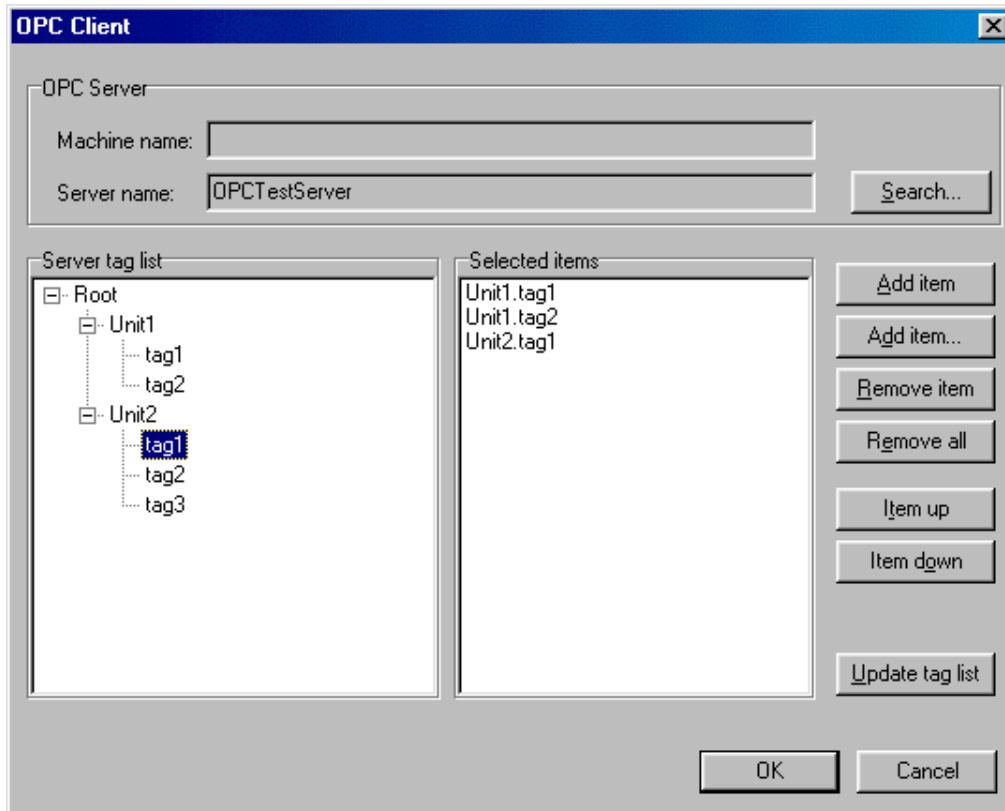
Die Parametrierung des *OPCServerOut*-Blocks ist völlig analog (siehe nachfolgende Bildschirmgrafik). Sämtliche Blockeingänge werden bei jedem Simulationsschritt aktualisiert (*UpdateTag*-Funktion der OPC-Schnittstelle).



Parameterdialog des OPCServerOut-Blocks

3. Betrieb von BORIS als OPC-Client

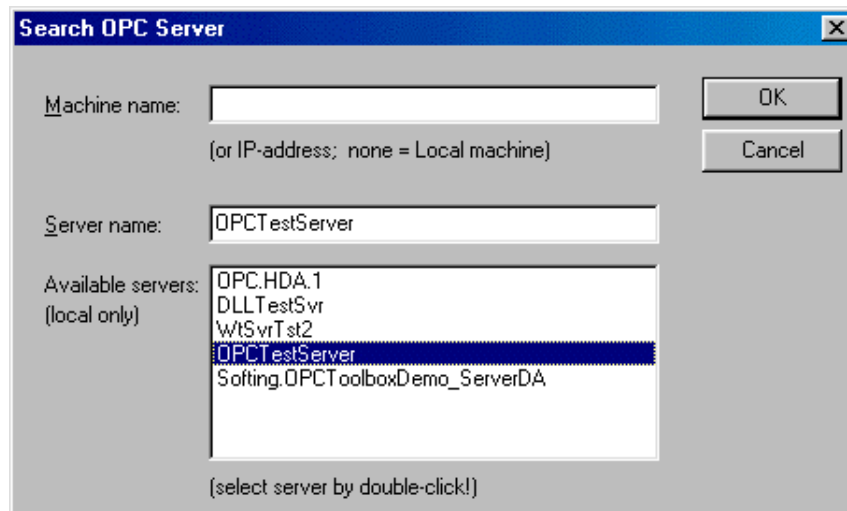
Mit Hilfe des *OPCClientIn*- bzw. *OPCClientOut*-Blocktyps kann BORIS Daten aus OPC-Servern lesen bzw. in diese schreiben (sofern die entsprechenden OPC-Items dort als *writable* definiert sind). Selbstverständlich können innerhalb einer BORIS-Struktur über mehrere *OPCClient*-Blöcke auch Verbindungen zu mehreren unterschiedlichen OPC-Servern gleichzeitig aufgebaut werden. Die OPC-Server können auf demselben Rechner wie BORIS (Betriebsart *local*) oder auf einem anderen Rechner (Betriebsart *remote*) installiert sein. Nachfolgende Bildschirmgrafik zeigt den *OPCClientIn*-Parameterdialog.



Parameterdialog des *OPCClientIn*-Blocks. Hier wurde als OPC-Server der BORIS-OPC-Testserver gewählt.

Über die Schaltfläche *Search...* wird zunächst der OPC-Server ausgewählt (siehe nachfolgende Bildschirmgrafik). Soll eine Verbindung zu einem auf einem anderen Rechner laufenden Server aufgebaut werden (Betriebsart *remote*), so ist der Name bzw. die IP-Adresse dieses Rechners im Eingabefeld *Machine name* anzugeben. Liegt der Server auf dem lokalen Rechner, muss dieser Eintrag leer bleiben.

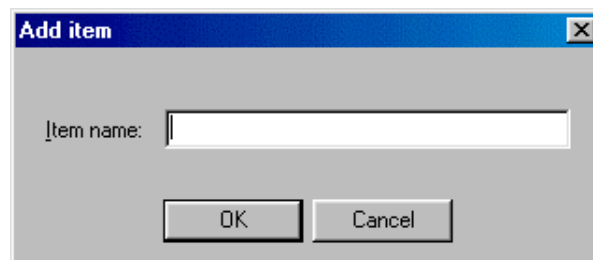
Im Eingabefeld *Server name* ist der Name des OPC-Servers anzugeben. Alle auf dem lokalen Rechner zur Verfügung stehenden OPC-Server werden im Listenfeld darunter angezeigt; der gewünschte Server (in untenstehendem Dialog der BORIS-OPC-Testserver *OPCTestServer*) kann hieraus einfach durch Doppelklick mit der linken Maustaste übernommen werden.



Wahl des OPC-Servers

Nach der Wahl des OPC-Servers versucht BORIS zunächst, eine temporäre Verbindung zum Server aufzubauen, um die verfügbaren Server-Tags auszulesen. Diese werden dann automatisch in einer Baumstruktur im Dialogfeld *Server tags* angezeigt. Über einen Doppelklick lassen sich nun die vom OPCClientIn-Blocks zu verarbeitenden Tags auswählen; sie erscheinen dann im Dialogfeld *Selected items*. Statt eines Doppelklicks lassen sich auch über die *Add item*-Schaltfläche einzelne Einträge hinzufügen. Die Anzahl der ausgewählten Tags bestimmt später automatisch die Anzahl der Blockausgänge.

Alternativ dazu (z. B., falls z. Z. keine Verbindung zum Server aufgebaut werden kann) können Items auch über die *Add item...*-Schaltfläche hinzugefügt werden. Ein Entfernen einzelner oder aller selektierten Items aus der Auswahlliste ist über die *Remove item-* bzw. *Remove all*-Schaltfläche möglich. Über die *Item up-* bzw. *Item down*-Schaltfläche kann die Reihenfolge der Items und damit ihre Zuordnung zu den Blockausgängen geändert werden.



Hinzufügen eines benutzerdefinierten Items

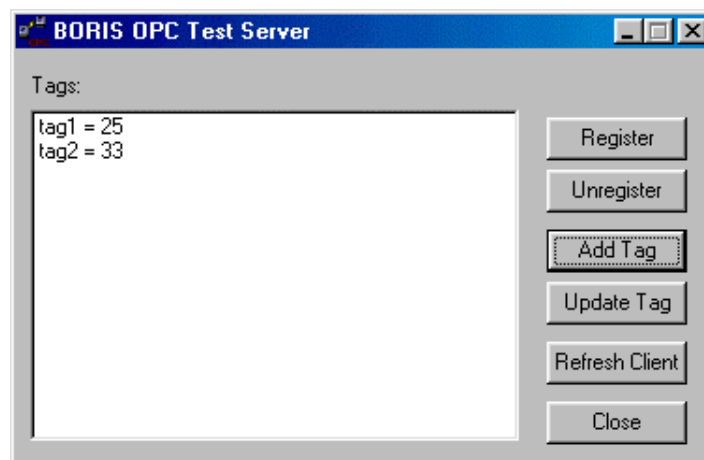
Wird der Parameterdialog des OPCClientIn-Blocks später erneut aufgerufen, baut BORIS *nicht* automatisch eine erneute Verbindung zum gewählten OPC-Server auf, sodass die *Server tag list* zunächst leer bleibt. Sie kann aber bei Bedarf über die *Update tag list*-Schaltfläche angefordert werden.

Die Verbindung zum gewählten OPC-Server wird zu Beginn der Simulation aufgebaut und nach Beendigung der Simulation wieder geschlossen. Außerhalb der Simulation besteht also keine Verbindung zum Server. Zu Beginn der Simulation werden alle Ausgänge des Blocks mit dem Wert -99999 vorbelegt. Dieser Wert wird dann solange beibehalten, bis vom entsprechenden OPC-Server für den entsprechenden Ausgang ein gültiger Wert gelesen werden konnte. BORIS interpretiert dabei die Fließkommazahlen-, Ganzzahl- und Booleschen Datentypen als Fließkommazahlen (der boolesche Wert TRUE wird als 1.0, der boolesche Wert FALSE als 0.0 übernommen); ein STRING-Datentyp wird als ungültig und demzufolge mit dem Wert -99999 interpretiert.

Die Parametrierung des *OPCClientOut*-Blocks ist völlig analog. Sämtliche Blockeingänge werden bei jedem Simulationsschritt aktualisiert (*WriteOPCItem*-Funktion der OPC-Schnittstelle). Dabei überprüft BORIS jedoch *nicht*, ob das entsprechende Item tatsächlich auch geschrieben werden kann, d. h. vom Server als *writable* deklariert wurde.

4. Der BORIS-OPC-Testserver

Für Testzwecke enthält die WinFACT-OPC-Toolbox einen einfachen Testserver (*OPCTESTSERVER.EXE*), der nach der Installation aus der WinFACT-Programmgruppe heraus gestartet werden kann.



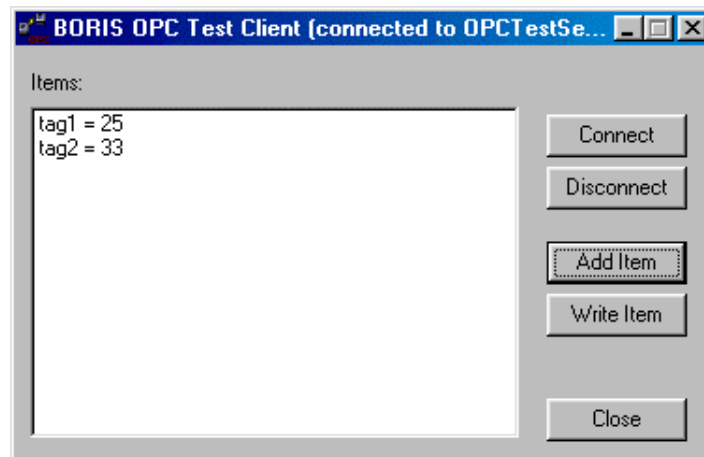
Der OPC-Testserver

Nach dem erstmaligen Starten muss der Server über die *Register*-Schaltfläche zunächst einmalig in der Windows-Registry angemeldet werden; der Server trägt sich dort dann mit der Bezeichnung *OPCTestServer* ein. Bei Bedarf kann der Registry-Eintrag über die *Unregister*-Schaltfläche jederzeit wieder entfernt werden.

Über die *Add Tag*-Schaltfläche lassen sich nun beliebige Server Tags mit ihrem Namen und ihrem Wert definieren (in obigem Dialog wurden die beiden Tags *tag1* und *tag2* definiert). Dabei wird sowohl das flache (FLAT) als auch das hierarchische (HIERARCHICAL) OPC-Namensmodell unterstützt. Über die *Update Tag*-Schaltfläche kann der Wert bereits vorhandener Tags geändert werden. Alle Tags werden als *writable* definiert; sie können also von einem OPC-Client überschrieben werden. Geänderte Werte werden automatisch an alle angemeldeten Clients weitergeleitet; bei Bedarf kann diese Aktualisierung auch über die *Refresh Client*-Schaltfläche explizit ausgelöst werden.

5. Der BORIS-OPC-Testclient

Zusätzlich zum Testserver enthält die Toolbox auch einen einfachen Testclient (*OPCTESTCLIENT.EXE*), der zum Testen beliebiger OPC-Server benutzt werden kann.



Der OPC-Testclient (hier verbunden mit dem OPC-Testserver)

Über die *Connect*-Schaltfläche wird zunächst der gewünschte OPC-Server ausgewählt (hier werden nur lokale Server unterstützt!). Die Verbindung wird dann im Titelbalken des Clients angezeigt und kann später über die *Disconnect*-Schaltfläche wieder geschlossen werden. Nach Aktivierung der Verbindung können über die *Add Item*-Schaltfläche die anzuzeigenden OPC-Items ausgewählt werden. Über die *Write Item*-Schaltfläche schließlich können (sofern vom Server zugelassen) neue Werte in die entsprechenden Items geschrieben werden.