

Description: Guide to Performance Optimization with DataWorX32 version 9.

OS Requirement: Win 2000 Pro/Server, XP Pro, Server 2003

General Requirement: General knowledge of DataWorX32

Loading the Database

When DataWorX32 Runtime starts, it needs to load the active configuration database. To do so, it may need to load components for accessing databases (OLE DB Provider and related DLLs, depending on what kind of database is used). Loading these components may be a time consuming process.

The first OPC client that connects to DataWorX32 may take longer to get the initial updates because DataWorX32 goes into runtime when the first client tries to connect to it when running as an application.

To speed up the process of connecting the first OPC client, it is recommended to pre-start DataWorX32 Runtime. You may auto-start DataWorX32 with GenTray, or run DataWorX32 as an automatically starting Service.

Using Registers

Using registers will improve the response time in communication with OPC servers. You can detect noticeable improvements when:

- Adding/removing items to the OPC servers is very slow, and
- The items on the OPC servers do not change very often.

When you use a register, the input OPC item is added on the DataWorX32 Runtime startup to maintain an open connection to the OPC server. By default, this connection is kept regardless of whether there is a client asking for data or not. This way, the registers keep their value up to date to simulate an immediate response. This means that DataWorX32 has to update the registers even if they are unused. Thus, we recommend you to use a dedicated machine for this solution.

To prevent DataWorX32 from polling registers even when clients are not connected, check the “Release inputs when not in use” option on the **Properties** tab of the appropriate register.

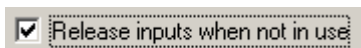


Figure 1 - Releasing inputs when not in use

Writing to OPC Items through Registers

You may need to set up a register to have its input and output connected to the same OPC item. In such a case, it is

recommended to disable input updates propagation with a checkbox on the **Properties** page of that register.

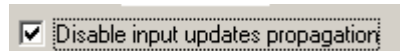


Figure 2 – Disabling input updates propagation

This setting causes the updates coming from the OPC item to not be written to the output (to the same OPC item). Only updates from outside of DataWorX32 are written to the outputs of such register. This prevents infinite loops that may happen when writing to the register faster, than the connected OPC item can handle.

Using OPC Item Cache in GenClient

For communication with OPC servers, DataWorX32 uses the GenClient.dll, as any other client application. Unlike other client applications, DataWorX32 is also an OPC server, allowing many OPC clients to connect to it. DataWorX32 will then get data from other OPC servers on behalf of the clients connected to it. Since each of the connected clients may switch displays, write to an OPC tag, log data, etc., the set of OPC items DataWorX32 reads from the servers may change quite often.

In such scenario, using the OPC Item Cache in GenClient may be helpful. To enable the cache, open the GenBroker Configurator by going to **Start → Programs → ICONICS Tools → GenBroker Configurator**. Click on edit once the application is open, and then expand **Advanced → Common → Optimization**, and check **OPC Item Cache**.



Figure 3 - Setting up OPC Item Cache in GenBroker Configurator