

# *DATAWORX32*

## Performance White Paper

---

AUGUST 2004

VERSION 8



*OPC Automation at your fingertips*

100 Foxborough Blvd., Foxborough, MA 02035  
508.543.8600 • FAX 508.543.1503  
[www.iconics.com](http://www.iconics.com) • E-mail [info@iconics.com](mailto:info@iconics.com)

© 2004 ICONICS, Inc. All Rights Reserved.  
Specifications and information subject to change without notice.

# Contents

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| <b>1</b> | <b>About This Document .....</b>                                      | <b>3</b>  |
| 1.1      | Scope of the Document .....                                           | 3         |
| 1.2      | Copyright and Confidentiality .....                                   | 3         |
| 1.3      | Revision History .....                                                | 3         |
| <b>2</b> | <b>Introduction .....</b>                                             | <b>4</b>  |
| 2.1      | Overview .....                                                        | 4         |
| <b>3</b> | <b>Test Setup .....</b>                                               | <b>6</b>  |
| 3.1      | Hardware .....                                                        | 6         |
| 3.1.1    | PC1: OPC Server .....                                                 | 6         |
| 3.1.2    | PC2: DataWorX32 Server .....                                          | 6         |
| 3.1.3    | PC3: Client 1 .....                                                   | 6         |
| 3.1.4    | PC4: Client 2 .....                                                   | 6         |
| 3.2      | Software .....                                                        | 6         |
| 3.2.1    | PC1: OPC Server .....                                                 | 6         |
| 3.2.2    | PC2: DataWorX32 Server .....                                          | 7         |
| 3.2.3    | PC3: Client 1 .....                                                   | 7         |
| 3.2.4    | PC4: Client 2 .....                                                   | 7         |
| 3.3      | Network Layout .....                                                  | 8         |
| 3.4      | Testing Methodology .....                                             | 9         |
| <b>4</b> | <b>Test Results .....</b>                                             | <b>10</b> |
| 4.1      | Test Case 1 DataWorX32 with 1000 Digital Registers .....              | 10        |
| 4.2      | Test Case 2 DataWorX32 with 5000 Digital Registers .....              | 11        |
| 4.3      | Test Case 3 DataWorX32 with 10,000 Digital Registers .....            | 12        |
| 4.4      | Test Case 4 DataWorX32 with 15,000 Digital Registers .....            | 13        |
| 4.5      | Test Case 5 DataWorX32 with 1200 Digital & Analog Registers .....     | 14        |
| 4.6      | Test Case 6 DataWorX32 with 6000 Digital & Analog Registers .....     | 15        |
| 4.7      | Test Case 7 DataWorX32 with 12,000 Digital & Analog Registers .....   | 16        |
| 4.8      | Test Case 8 DataWorX32 with 17,000 Digital & Analog Registers .....   | 17        |
| 4.9      | DataWorX32 Performance with 2 clients .....                           | 18        |
| 4.10     | Are as many updates received as there are sent from the server? ..... | 19        |
| 4.11     | Test Case 11 DataWorX32 with 1000 Digital Registers .....             | 21        |
| 4.12     | Test Case12 DataWorX32 with 5000 Digital Registers .....              | 22        |

|             |                                                                                 |           |
|-------------|---------------------------------------------------------------------------------|-----------|
| <b>4.13</b> | <b>Test Case 13 DataWorX32 with 10,000 Digital Registers .....</b>              | <b>23</b> |
| <b>4.14</b> | <b>Test Case14 DataWorX32 with 15,000 Digital Registers .....</b>               | <b>24</b> |
| <b>4.15</b> | <b>Test Case 15 DataWorX32 with 1200 Digital &amp; Analog Registers .....</b>   | <b>25</b> |
| <b>4.16</b> | <b>Test Case 16 DataWorX32 with 6000 Digital &amp; Analog Registers .....</b>   | <b>26</b> |
| <b>4.17</b> | <b>Test Case 17 DataWorX32 with 12,000 Digital &amp; Analog Registers .....</b> | <b>27</b> |
| <b>4.18</b> | <b>Test Case 18 DataWorX32 with 17,000 Digital &amp; Analog Registers .....</b> | <b>28</b> |
| <b>4.19</b> | <b>DataWorX32 Performance with 2 clients .....</b>                              | <b>29</b> |
| <b>4.20</b> | <b>Are as many updates received as there are sent from the server? .....</b>    | <b>30</b> |
| <b>5</b>    | <b>Conclusions .....</b>                                                        | <b>32</b> |

# **1 About This Document**

---

## ***1.1 Scope of the Document***

This document contains the performance test results for DataWorX32. The intended audience includes the design engineers working on the product, sales and marketing management as well as key customers providing input as to the features of this product.

## ***1.2 Copyright and Confidentiality***

The document is © Copyright 2004, ICONICS, Inc., Foxborough, Massachusetts as an unpublished work. It is considered CONFIDENTIAL, and contains trade secret information proprietary to ICONICS, Inc.

## ***1.3 Revision History***

Version 8.0 – Madhavi Mukkavilli, May 2004

Version 8.01 – RLA, Aug 2004

## 2 Introduction

---

### 2.1 Overview

The intention for this document is to summarize the performance test results for DataWorX32 V8.0

In this test, we are

- ✓ Measuring the CPU, Memory Usage and Virtual Memory Usage by DataWorX32.
- ✓ We compare the usage for Small, Medium and Large-Scale applications.
- ✓ We also compare the usage between Genesis V 7.1 and Genesis V8, and also for Small, Medium and Large-Scale Applications.

Small, Medium and Large Scale Applications are defined below:

#### Only Digital Registers:

- Small Application - DataWorX32 has 1000 Registers: Of these 100 Registers change every 1-second, 400 changing every 10 seconds, and 500 change once a minute.
- Medium Application - DataWorX32 has 5000 Registers: Of these 500 Registers change every 1-second, 2000 changing every 10 seconds, and 2500 change once a minute
- Large Application - DataWorX32 has 10,000 Registers: Of these 1000 Registers change every 1-second, 4000 changing every 10 seconds, and 5000 change once a minute
- Extra Large Application - DataWorX32 has 15,000 Registers: Of these 1000 Registers change every 1-second, 4000 changing every 10 seconds, and 10,000 change once a minute

#### Analog and Digital Registers:

- Small Application - DataWorX32 has 1200 Registers: Of these 100 Registers change every 1-second, 400 changing every 10 seconds, and 500 change once a minute. All these are Digitals. The remaining 200 Registers are Analog of which 100 change every 1 second, and the other 100 change twice a second.
- Medium Application - DataWorX32 has 5000 Registers: Of these 500 Registers change every 1-second, 2000 changing every 10 seconds, and 2500 change once a minute. All these are Digitals. The remaining 1000 Registers are Analog of which 500 change every 1 second, and the other 500 change twice a second.

- Large Application - DataWorX32 has 12,000 Registers: Of these 1000 Registers change every 1-second, 4000 changing every 10 seconds, and 5000 change once a minute. All these are Digitals. The remaining 2000 Registers are Analog of which 1000 change every 1 second, and the other 1000 change twice a second.
- Extra Large Application - DataWorX32 has 17,000 Registers: Of these 1000 Registers change every 1-second, 4000 changing every 10 seconds, and 10,000 change once a minute The remaining 2000 Registers are Analog of which 1000 change every 1 second, and the other 1000 change twice a second.

## 3 Test Setup

---

### 3.1 Hardware

#### 3.1.1 PC1: OPC Server

OS: Windows XP, Service Pack 1  
RAM: 512 MB  
Processor: Pentium 4, CPU 1.79 GHz

#### 3.1.2 PC2: DataWorX32 Server

OS: Windows XP, Service Pack 1  
RAM: 504 MB  
Processor: Pentium 4, CPU 2.50 GHz

#### 3.1.3 PC3: Client 1

OS: Windows 2000, Service Pack 4  
RAM: 260 MB  
Processor: Pentium 4, CPU 1.80 GHz

#### 3.1.4 PC4: Client 2

OS: Windows 2000, Service Pack 4  
RAM: 130 MB  
Processor: Pentium 2, CPU 400 MHz

### 3.2 Software

#### 3.2.1 PC1: OPC Server

ICONICS Simulator 3.05

For DataWorX32 V 7.1 Testing the following mix of data tags were used:

**Boolean Tags:**

G01-G05: 1000 Tags each group changing every 1-second  
G06-G10: 1000 Tags each group changing every 10 seconds  
G11-G15: 1000 Tags each group changing every 1-minute  
G16: 5000 Tags changing every 1-minute  
G19, G20: 1000 Tags each group changing every 1-second

**Analog Tags:**

G17: 1000 tags changing every 5000 msec

G18: 1000 tags changing every 1 sec

### 3.2.2 PC2: DataWorX32 Server

For V 7.1 Test: Genesis V 7.1 and

For V8: Genesis V8 Beta IV

### 3.2.3 PC3: Client 1

For V 7.1 Test: Genesis V 7.1 and

For V8: Genesis V8 Beta IV

### 3.2.4 PC4: Client 2

For V 7.1 Test: Genesis V 7.1 and

For V8: Genesis V8 Beta IV

### 3.3 Network Layout

The following diagram gives an idea of the network setup. These tests were run during normal office hours and so take into account normal network load:

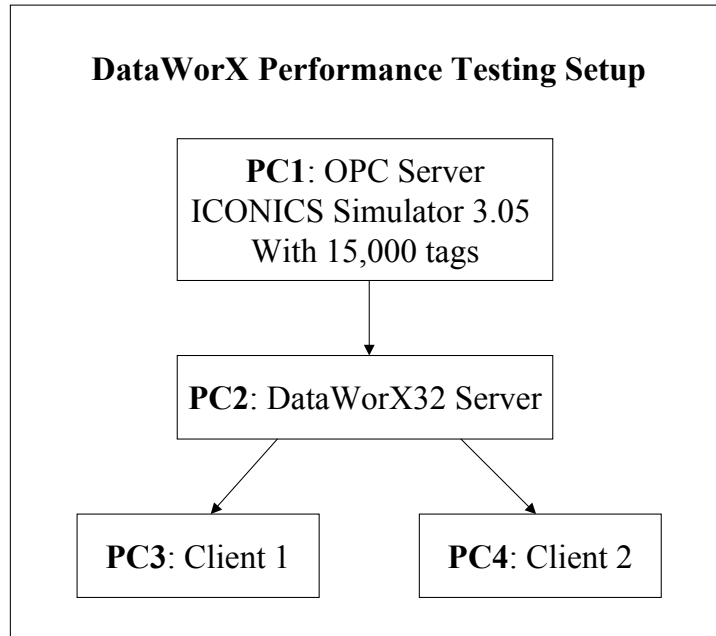


**Figure 3.3**

### 3.4 Testing Methodology

The client machines have GraphWorX32 Displays running on them. On one client, we have GraphWorX32 display static in runtime. On the other machine, we have GraphWorX32 running 10 displays with screen switching. Carousel is used for screen switching purpose.

8 different tests were run for measuring DataWorX32 Performance. Each time DataWorX32 had a different configuration file with different number of Registers. GraphWorX32 on one of the client machines is static when put into runtime, and the other GraphWorX32 client has Screen Switching when put into runtime. The CPU, Memory and Virtual Memory Usage were measured for DataWorX32 on the DataWorX32 server machine.

First start DataWorX32 and let it run for 15 minutes before the values are taken. The CPU usage was taken from Performance Monitor, and Virtual Memory Usage was taken from the Task Manager. After 15 minutes the first GraphWorX32 client is put in to runtime. Wait for 15 minutes before the values are taken down. Then put the second client in runtime and wait for 15 minutes before the values are taken down (This gives values for 2 clients). Then close the first client, and wait for 15 minutes before noting down the values.

**DCOM** was the communication channel for all the tests described below.

For the GraphWorX32 displays, the screens were created using color dynamics and process points, the reason being that color dynamics require the least amount of client side rendering, while process points require the most.

First Genesis V 7.1 was loaded on the machines and the following tests were run. After the tests are completed, the machines were ghosted and Genesis V8, Beta IV was installed on them. The same tests were run again, with the same sets of files used in 7.1. The GraphWorX32 displays are re-used for testing in V8.

## 4 Test Results

### 4.1 Test Case 1 DataWorX32 with 1000 Digital Registers

DataWorX32 has 1000 Registers in it, each changing every second.

GraphWorX32 Static Display: 1000 tags with 100 changing every 1-second, 400 changing every 10 seconds, 500 changing every 1 minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 10 tags changing every 1-second, 40 tags changing once every 10 seconds, and 50 tags changing once every 1 minute

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.2   | 10.76                       |
| 1 Client – GraphWorX32 Static           | 0.21  | 10.81                       |
| 1 Client – GraphWorX32 Screen Switching | 0.23  | 10.82                       |
| 2 Clients                               | 0.24  | 10.83                       |

**Table 4.1**

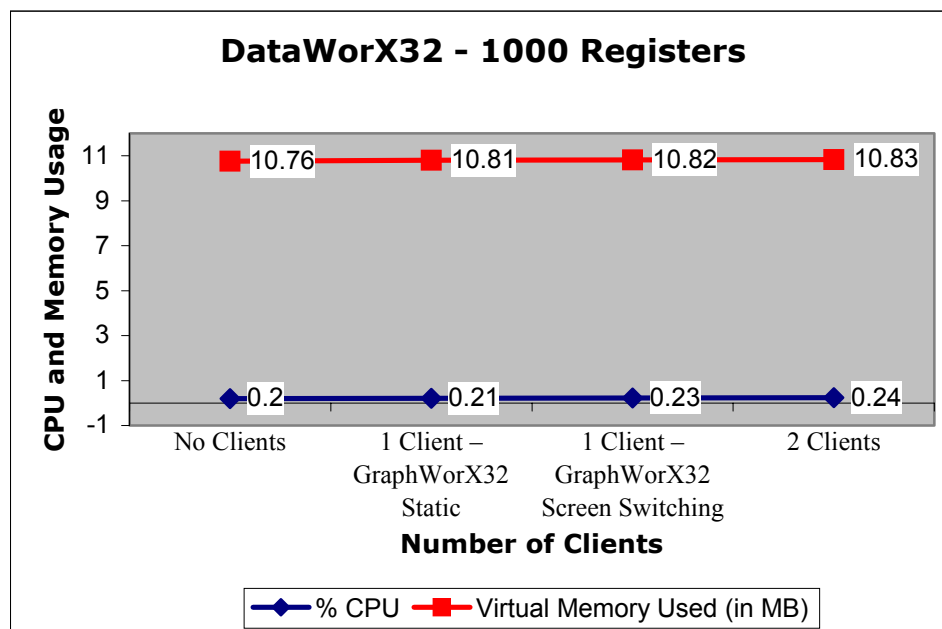


**Chart 4.1**

## 4.2 Test Case 2 DataWorX32 with 5000 Digital Registers

2 GraphWorX32 Clients requesting data from DataWorX32. DataWorX32 has 5000 Registers

GraphWorX32 Static Display: 5000 tags with 500 changing every 1-second, 2000 changing every 10 seconds, 2500 changing every 1-minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 50 tags changing every 1-second, 200 tags changing once every 10 seconds, and 250 tags changing once every 1 minute

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.24  | 19.41                       |
| 1 Client – GraphWorX32 Static           | 0.9   | 19.87                       |
| 1 Client – GraphWorX32 Screen Switching | 0.37  | 19.86                       |
| 2 Clients                               | 1.2   | 20.02                       |

**Table 4.2**

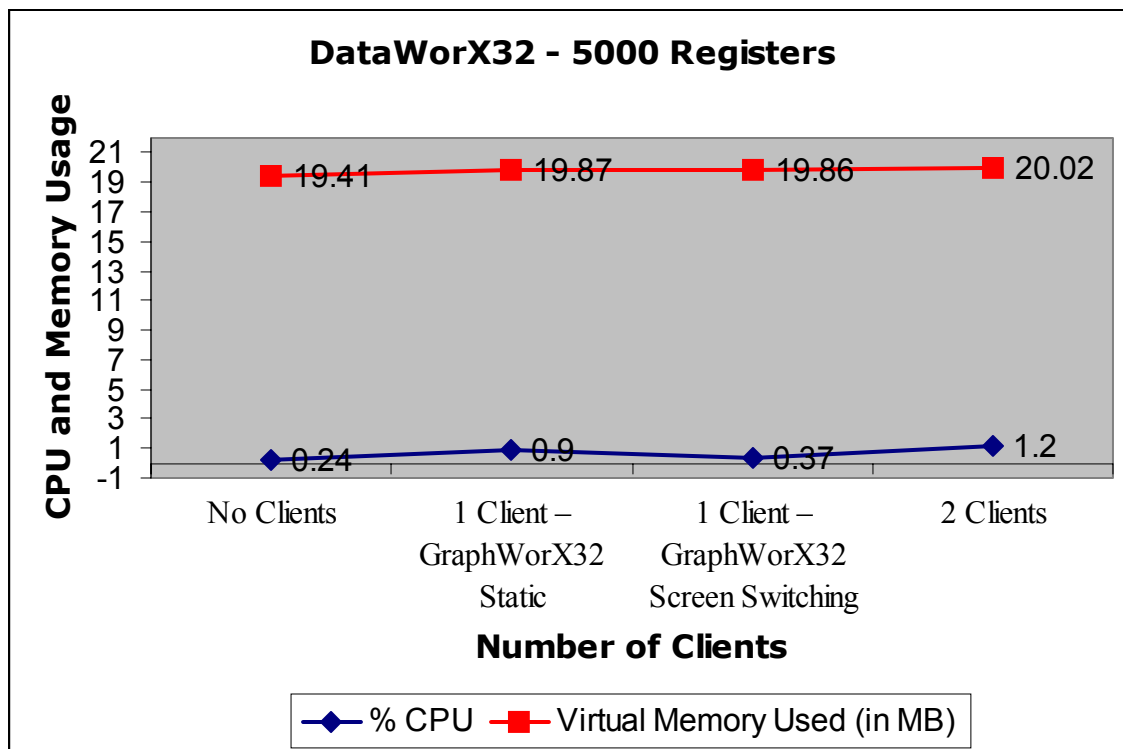


**Chart 4.2**

### 4.3 Test Case 3 DataWorX32 with 10,000 Digital Registers

2 GraphWorX32 Clients requesting data from DataWorX32. DataWorX32 has 10,000 Registers

GraphWorX32 Static Display: 1000 changing every 1-second, 4000 changing every 10 seconds, 5000 changing every 1-minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 100 tags changing every 1-second, 400 tags changing once every 10 seconds, and 500 tags changing once every 1 minute

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.99  | 34.48                       |
| 1 Client – GraphWorX32 Static           | 6.5   | 36.56                       |
| 1 Client – GraphWorX32 Screen Switching | 1.15  | 34.84                       |
| 2 Clients                               | 6.7   | 36.65                       |

**Table 4.3**

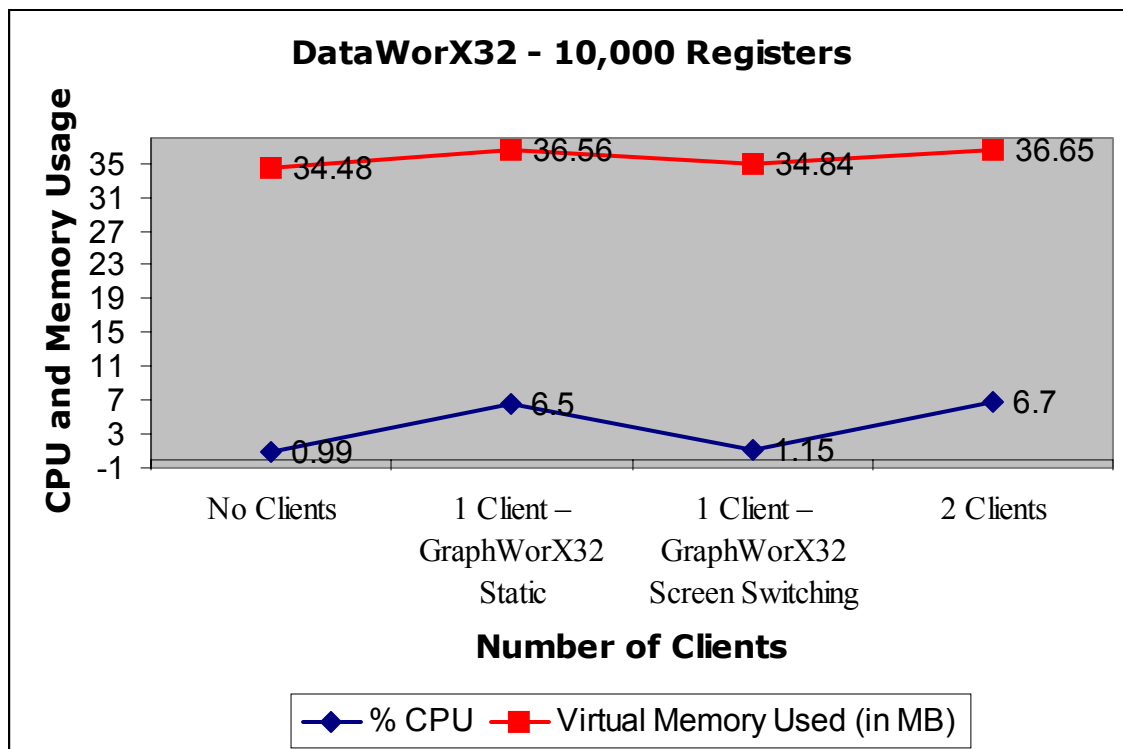


**Chart 4.3**

#### 4.4 Test Case 4 DataWorX32 with 15,000 Digital Registers

2 GraphWorX32 Clients requesting data from DataWorX32. DataWorX32 has 15,000 Registers

GraphWorX32 Static Display: 1000 changing every 1-second, 4000 changing every 10 seconds, 10,000 changing every 1-minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 100 tags changing every 1-second, 400 tags changing once every 10 seconds, and 500 tags changing once every 1 minute

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 1.08  | 47.41                       |
| 1 Client – GraphWorX32 Static           | 7.6   | 48.21                       |
| 1 Client – GraphWorX32 Screen Switching | 1.18  | 48.69                       |
| 2 Clients                               | 7.9   | 49.36                       |

**Table 4.4**

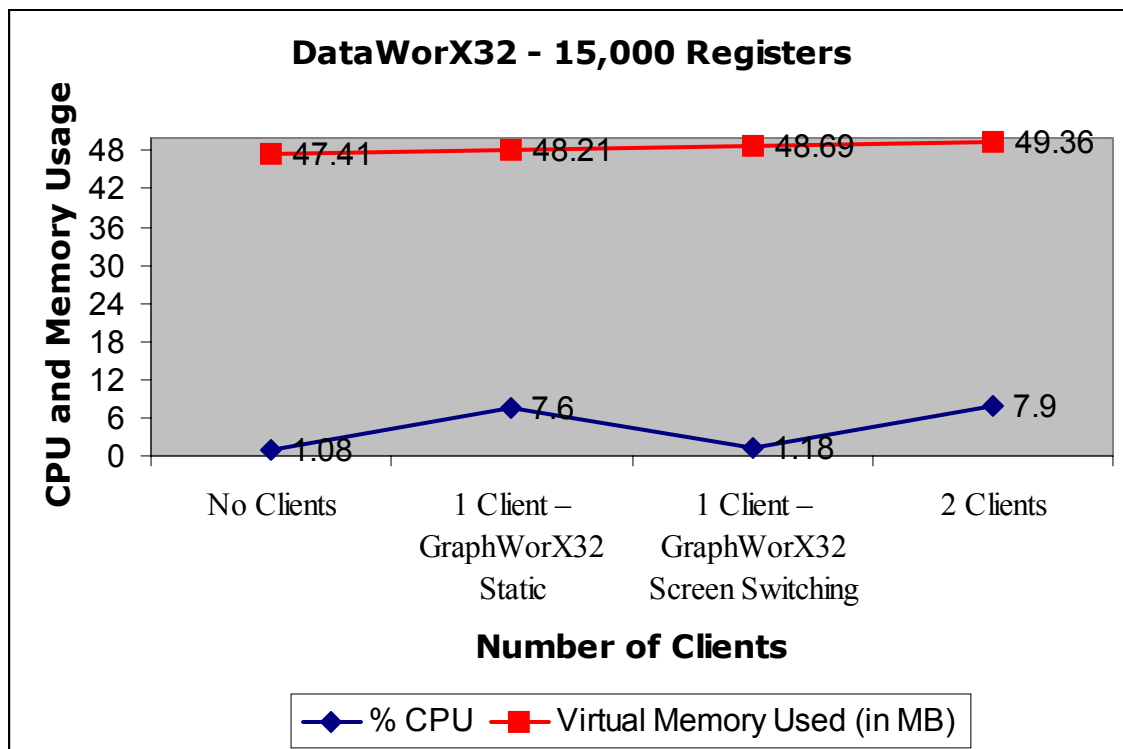


**Chart 4.4**

## 4.5 Test Case 5 DataWorX32 with 1200 Digital & Analog Registers

DataWorX32 has 1200 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 1200 Digital Registers with 100 changing every 1-second, 400 changing every 10 seconds, 500 changing every 1 minute. The remaining 200 are Analog Registers of which 100 change every 1 second, and 100 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) displays have 10 Registers changing every 1-second, 40 Registers changing once every 10 seconds, and 50 Registers changing once every 1 minute. The other 2 displays have 100 Registers each, which are Analog Registers. Of which 50 change once a second and 50 change twice every 1 second

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.22  | 11.41                       |
| 1 Client – GraphWorX32 Static           | 0.22  | 11.51                       |
| 1 Client – GraphWorX32 Screen Switching | 0.18  | 11.50                       |
| 2 Clients                               | 0.3   | 11.52                       |

**Table 4.5**

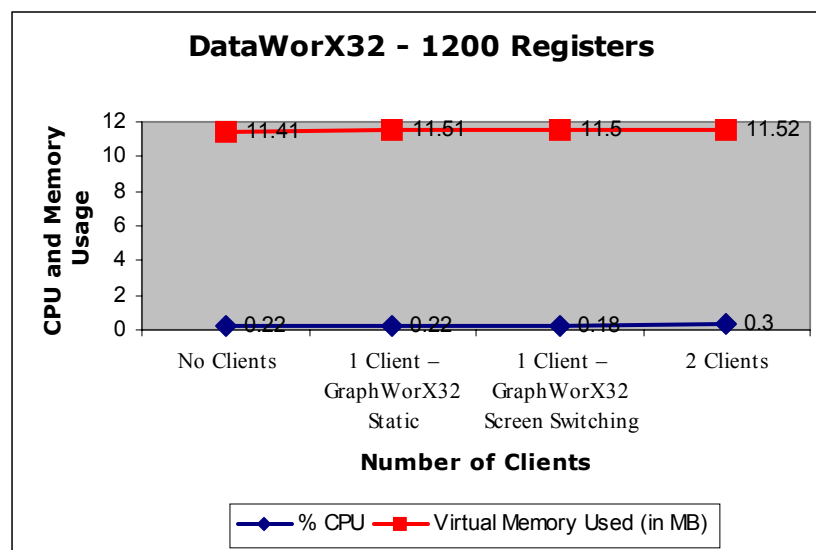


**Chart 4.5**

## 4.6 Test Case 6 DataWorX32 with 6000 Digital & Analog Registers

DataWorX32 has 6000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 6000 Digital Registers with 500 changing every 1-second, 2000 changing every 10 seconds, 5000 changing every 1 minute. The remaining 1000 are Analog Registers of which 500 change every 1 second, and 500 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) display has 50 Registers changing every 1-second, 200 Registers changing once every 10 seconds, and 250 Registers changing once every 1 minute. The other 2 displays have 500 Registers each, which are Analog Registers. Of which 250 change once a second and 250 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 1.29  | 21.80                       |
| 1 Client – GraphWorX32 Static           | 3.5   | 22.04                       |
| 1 Client – GraphWorX32 Screen Switching | 1.4   | 22.52                       |
| 2 Clients                               | 3.6   | 22.55                       |

**Table 4.6**

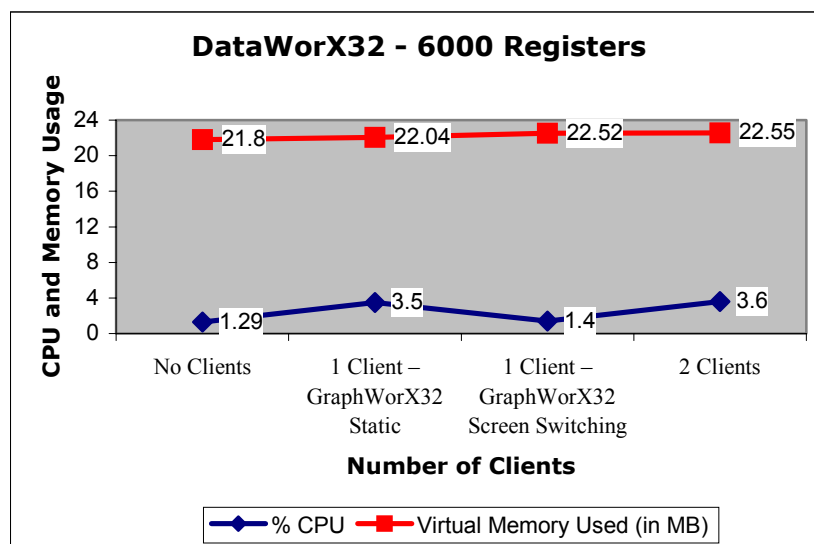


**Chart 4.6**

## 4.7 Test Case 7 DataWorX32 with 12,000 Digital & Analog Registers

DataWorX32 has 12,000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 12,000 Digital Registers with 500 changing every 1-second, 2000 changing every 10 seconds, 5000 changing every 1 minute. The remaining 2000 are Analog Registers of which 1000 change every 1 second, and 1000 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) display has 100 Registers changing every 1-second, 400 Registers changing once every 10 seconds, and 500 Registers changing once every 1 minute. The other 2 displays have 1000 Registers each, which are Analog Registers. Of which 500 change once a second and 500 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 4.4   | 41.19                       |
| 1 Client – GraphWorX32 Static           | 13.6  | 41.23                       |
| 1 Client – GraphWorX32 Screen Switching | 4.7   | 41.46                       |
| 2 Clients                               | 14.34 | 41.59                       |

**Table 4.7**

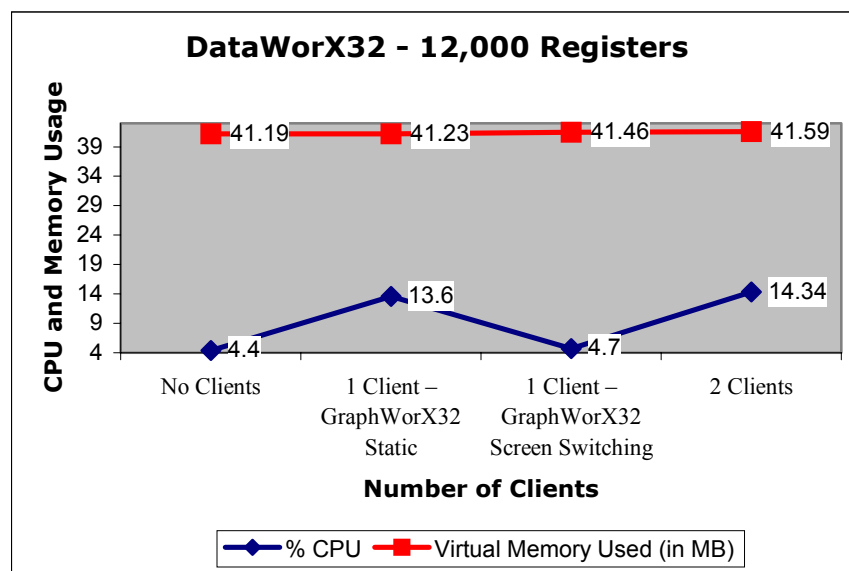


**Chart 4.7**

## 4.8 Test Case 8 DataWorX32 with 17,000 Digital & Analog Registers

DataWorX32 has 17,000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 12,000 Digital Registers with 500 changing every 1-second, 2000 changing every 10 seconds, 5000 changing every 1 minute. The remaining 2000 are Analog Registers of which 1000 change every 1 second, and 1000 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) display has 100 Registers changing every 1-second, 400 Registers changing once every 10 seconds, and 500 Registers changing once every 1 minute. The other 2 displays have 1000 Registers each, which are Analog Registers. Of which 500 change once a second and 500 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 4.2   | 55.32                       |
| 1 Client – GraphWorX32 Static           | 6.7   | 56.05                       |
| 1 Client – GraphWorX32 Screen Switching | 4.2   | 56.24                       |
| 2 Clients                               | 7.9   | 56.70                       |

**Table 4.8**

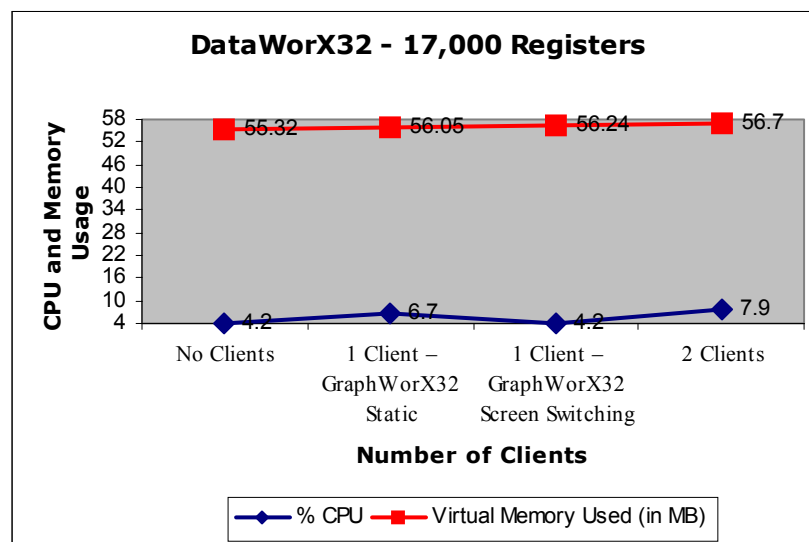


**Chart 4.8**

## 4.9 DataWorX32 Performance with 2 clients

The following chart shows the DataWorX32 performance as the numbers of tags are increased. The values are when 2 clients are requesting data from DataWorX32

| Number of Clients | % CPU | Virtual Memory Used (in MB) |
|-------------------|-------|-----------------------------|
| 1000 Registers    | 0.24  | 10.83                       |
| 1200 Registers    | 0.3   | 11.52                       |
| 5000 Registers    | 1.2   | 20.02                       |
| 6000 Registers    | 3.6   | 22.55                       |
| 10,000 Registers  | 6.7   | 36.65                       |
| 12,000 Registers  | 14.34 | 41.59                       |
| 15,000 Registers  | 7.9   | 49.36                       |
| 17,000 Registers  | 7.9   | 56.70                       |

**Table 4.9**

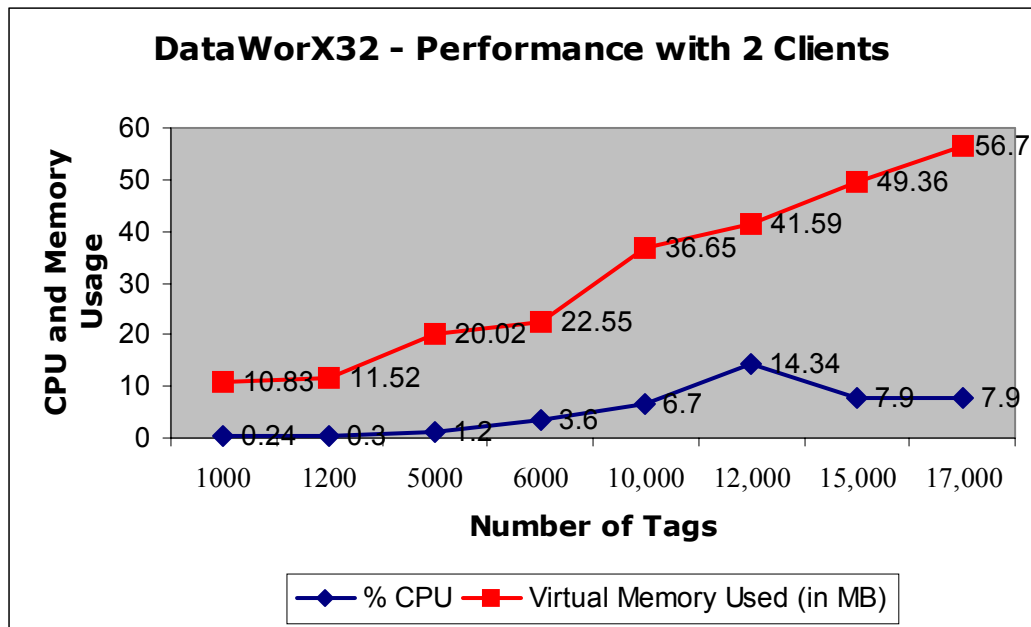
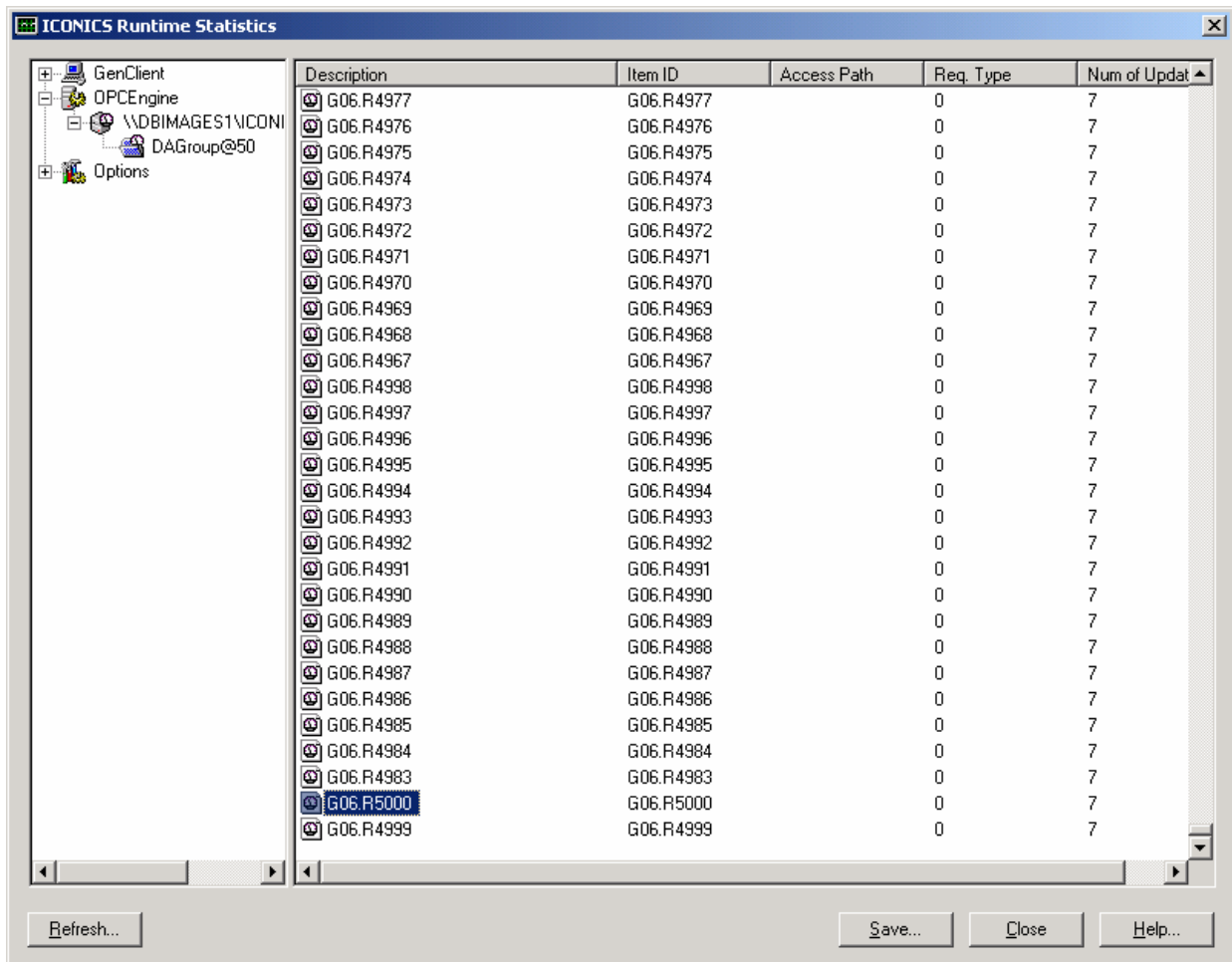


**Chart 4.9**

#### 4.10 Are as many updates received as there are sent from the server?

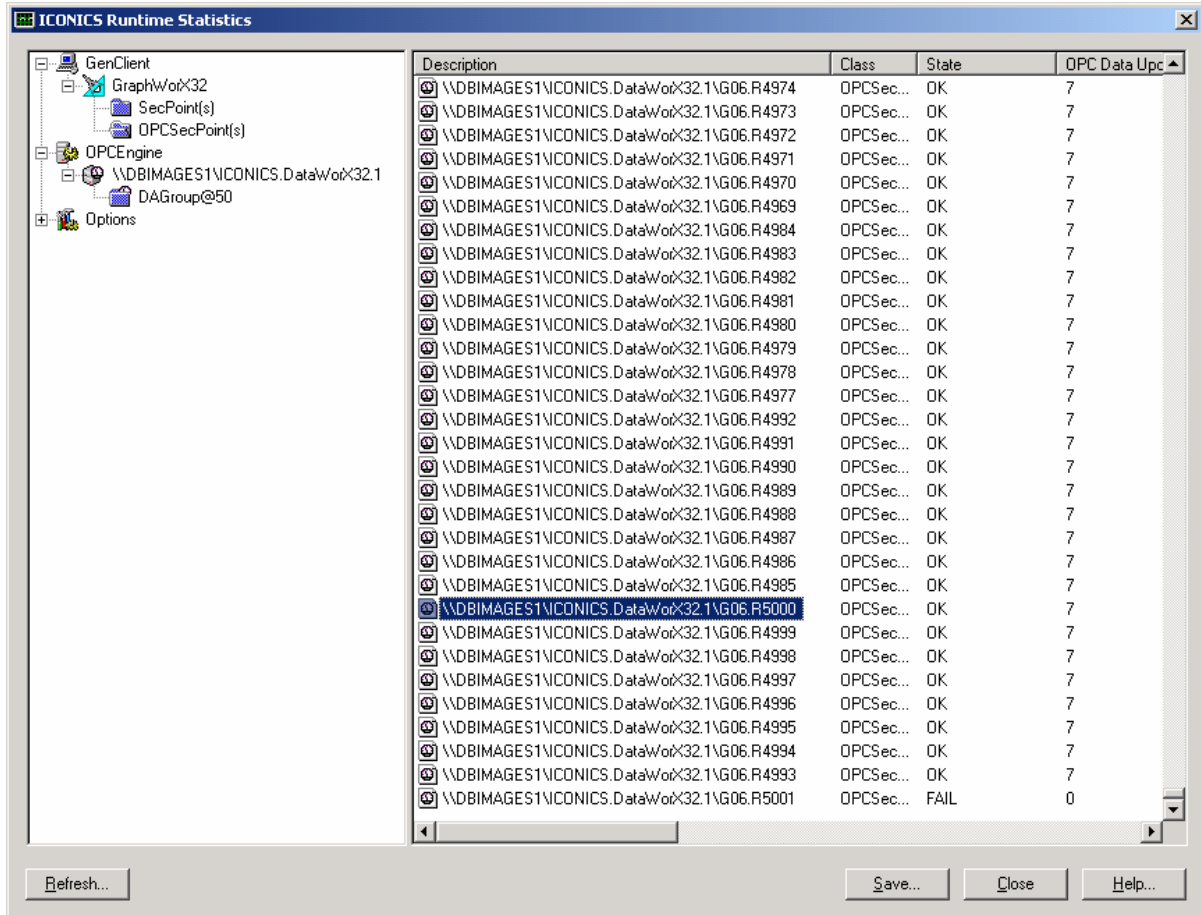
This test can be done by checking the GenStatistics on both the server-side and client-side. What the statistics files should show is there are as many updates received as there are sent. While the files cannot be saved at exactly the same time, the figures can be slightly off. The figures below show the results of one of the tests. As can be seen the number of updates sent matches the number of updates received. (The screens show the data for Test 2 and Test 3)

For Test 3, the first update came in 23 seconds.



| Description | Item ID   | Access Path | Req. Type | Num of Updat |
|-------------|-----------|-------------|-----------|--------------|
| G06.R4977   | G06.R4977 |             | 0         | 7            |
| G06.R4976   | G06.R4976 |             | 0         | 7            |
| G06.R4975   | G06.R4975 |             | 0         | 7            |
| G06.R4974   | G06.R4974 |             | 0         | 7            |
| G06.R4973   | G06.R4973 |             | 0         | 7            |
| G06.R4972   | G06.R4972 |             | 0         | 7            |
| G06.R4971   | G06.R4971 |             | 0         | 7            |
| G06.R4970   | G06.R4970 |             | 0         | 7            |
| G06.R4969   | G06.R4969 |             | 0         | 7            |
| G06.R4968   | G06.R4968 |             | 0         | 7            |
| G06.R4967   | G06.R4967 |             | 0         | 7            |
| G06.R4998   | G06.R4998 |             | 0         | 7            |
| G06.R4997   | G06.R4997 |             | 0         | 7            |
| G06.R4996   | G06.R4996 |             | 0         | 7            |
| G06.R4995   | G06.R4995 |             | 0         | 7            |
| G06.R4994   | G06.R4994 |             | 0         | 7            |
| G06.R4993   | G06.R4993 |             | 0         | 7            |
| G06.R4992   | G06.R4992 |             | 0         | 7            |
| G06.R4991   | G06.R4991 |             | 0         | 7            |
| G06.R4990   | G06.R4990 |             | 0         | 7            |
| G06.R4989   | G06.R4989 |             | 0         | 7            |
| G06.R4988   | G06.R4988 |             | 0         | 7            |
| G06.R4987   | G06.R4987 |             | 0         | 7            |
| G06.R4986   | G06.R4986 |             | 0         | 7            |
| G06.R4985   | G06.R4985 |             | 0         | 7            |
| G06.R4984   | G06.R4984 |             | 0         | 7            |
| G06.R4983   | G06.R4983 |             | 0         | 7            |
| G06.R5000   | G06.R5000 |             | 0         | 7            |
| G06.R4999   | G06.R4999 |             | 0         | 7            |

Figure 4.10 a



| Description                                  | Class     | State | OPC Data Upc |
|----------------------------------------------|-----------|-------|--------------|
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4974 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4973 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4972 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4971 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4970 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4969 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4984 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4983 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4982 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4981 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4980 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4979 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4978 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4977 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4992 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4991 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4990 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4989 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4988 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4987 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4986 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4985 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R5000 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4999 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4998 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4997 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4996 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4995 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4994 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R4993 | OPCSec... | OK    | 7            |
| \\DBIMAGES1\\ICONICS.DataWorX32.1\\G06.R5001 | OPCSec... | FAIL  | 0            |

Refresh... Save... Close Help...

Figure 4.10 b

### 4.11 Test Case 11 DataWorX32 with 1000 Digital Registers

DataWorX32 has 1000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 1000 tags with 100 changing every 1-second, 400 changing every 10 seconds, 500 changing every 1 minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 10 tags changing every 1-second, 40 tags changing once every 10 seconds, and 50 tags changing once every 1 minute.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.02  | 11.20                       |
| 1 Client – GraphWorX32 Static           | 0.04  | 13.91                       |
| 1 Client – GraphWorX32 Screen Switching | 0.03  | 11.64                       |
| 2 Clients                               | 0.09  | 14.10                       |

**Table 4.11**

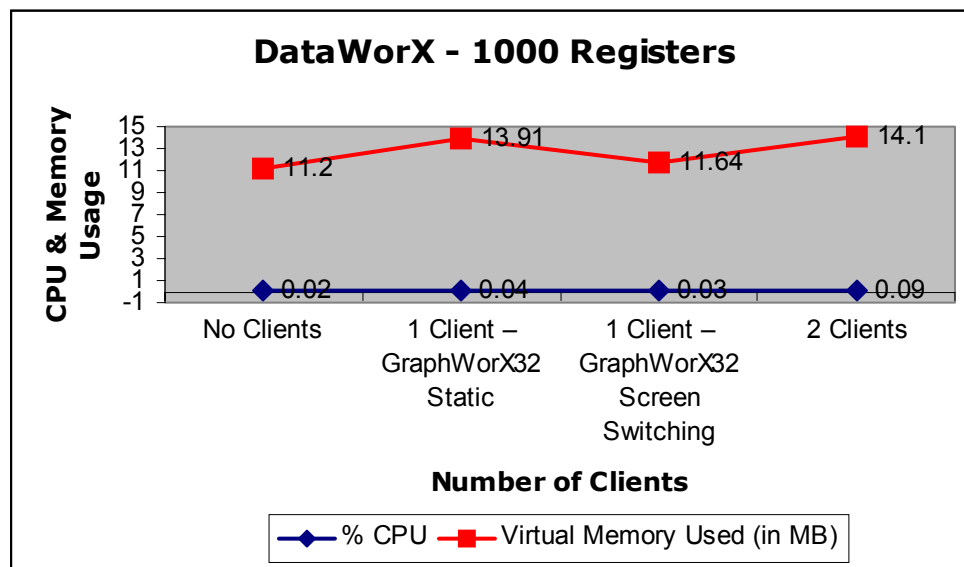


**Chart 4.11**

## 4.12 Test Case12 DataWorX32 with 5000 Digital Registers

2 GraphWorX32 Clients requesting data from DataWorX32. DataWorX32 has 5000 Registers

GraphWorX32 Static Display: 5000 tags with 500 changing every 1-second, 2000 changing every 10 seconds, 2500 changing every 1-minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 50 tags changing every 1-second, 200 tags changing once every 10 seconds, and 250 tags changing once every 1 minute.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.4   | 26.81                       |
| 1 Client – GraphWorX32 Static           | 1.5   | 45.08                       |
| 1 Client – GraphWorX32 Screen Switching | 0.7   | 28.95                       |
| 2 Clients                               | 1.6   | 45.52                       |

**Table 4.12**

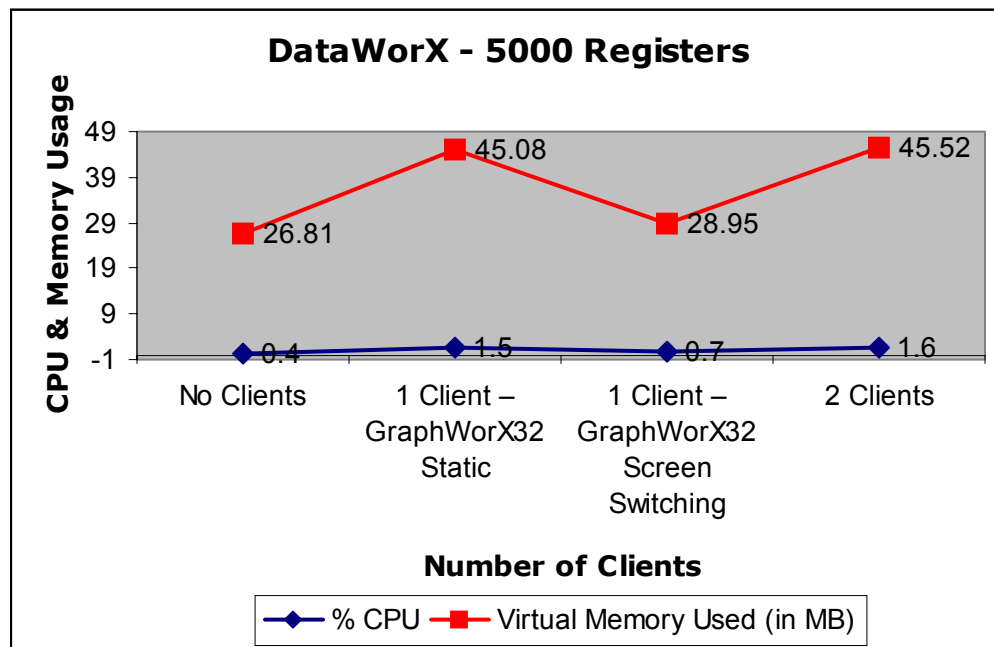


**Chart 4.12**

### 4.13 Test Case 13 DataWorX32 with 10,000 Digital Registers

2 GraphWorX32 Clients requesting data from DataWorX32. DataWorX32 has 10,000 Registers

GraphWorX32 Static Display: 1000 changing every 1-second, 4000 changing every 10 seconds, 5000 changing every 1-minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 100 tags changing every 1-second, 400 tags changing once every 10 seconds, and 500 tags changing once every 1 minute.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.9   | 47.92                       |
| 1 Client – GraphWorX32 Static           | 5.1   | 83.33                       |
| 1 Client – GraphWorX32 Screen Switching | 2.0   | 51.56                       |
| 2 Clients                               | 5.3   | 84.47                       |

**Table 4.13**

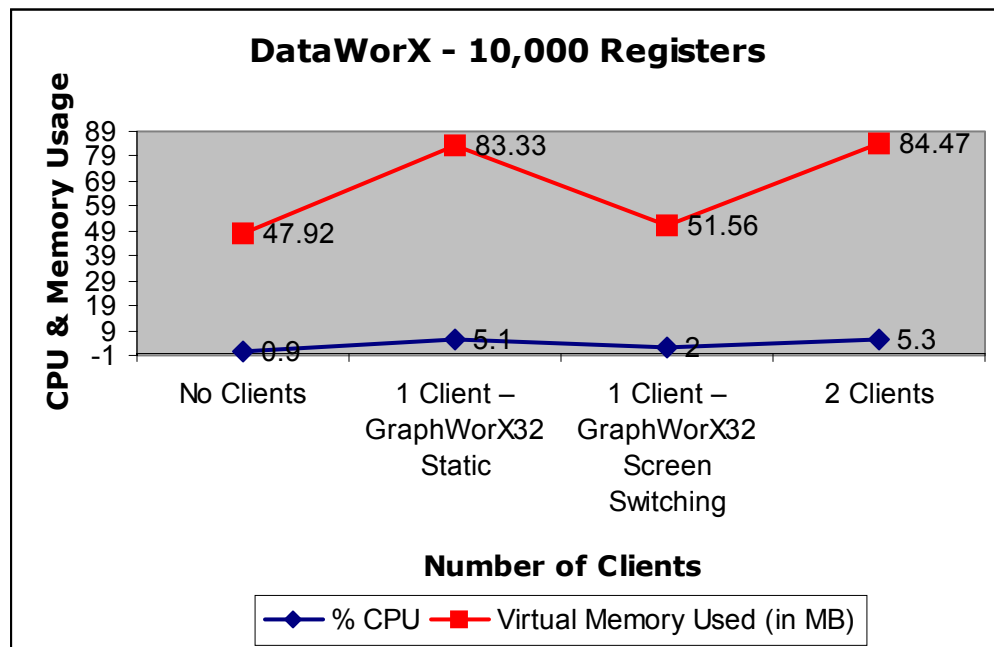


**Chart 4.13**

#### 4.14 Test Case14 DataWorX32 with 15,000 Digital Registers

2 GraphWorX32 Clients requesting data from DataWorX32. DataWorX32 has 15,000 Registers

GraphWorX32 Static Display: 1000 changing every 1-second, 4000 changing every 10 seconds, 10,000 changing every 1-minute

GraphWorX32 Screen Switches: Each GraphWorX32 display has 100 tags changing every 1-second, 400 tags changing once every 10 seconds, and 500 tags changing once every 1 minute.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 1.9   | 67.71                       |
| 1 Client – GraphWorX32 Static           | 5.2   | 103.03                      |
| 1 Client – GraphWorX32 Screen Switching | 2.2   | 71.78                       |
| 2 Clients                               | 5.7   | 104.12                      |

**Table 4.14**

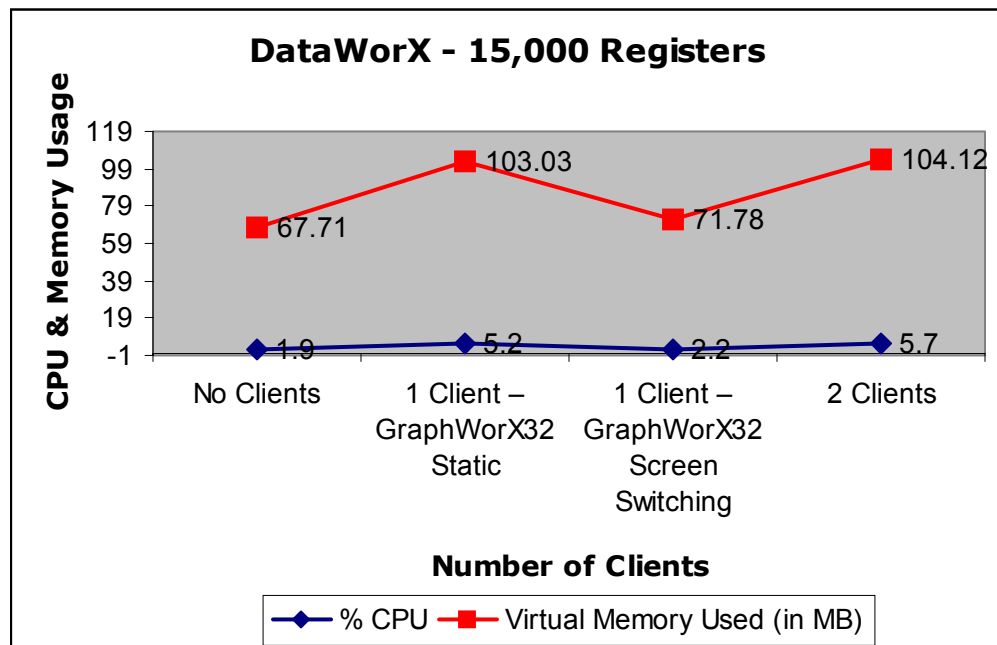


**Chart 4.14**

### 4.15 Test Case 15 DataWorX32 with 1200 Digital & Analog Registers

DataWorX32 has 1200 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 1200 Digital Registers with 100 changing every 1-second, 400 changing every 10 seconds, 500 changing every 1 minute. The remaining 200 are Analog Registers of which 100 change every 1 second, and 100 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) displays have 10 Registers changing every 1-second, 40 Registers changing once every 10 seconds, and 50 Registers changing once every 1 minute. The other 2 displays have 100 Registers each, which are Analog Registers. Of which 50 change once a second and 50 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 0.2   | 10.96                       |
| 1 Client – GraphWorX32 Static           | 0.3   | 15.50                       |
| 1 Client – GraphWorX32 Screen Switching | 0.06  | 12.10                       |
| 2 Clients                               | 0.3   | 15.77                       |

**Table 4.15**

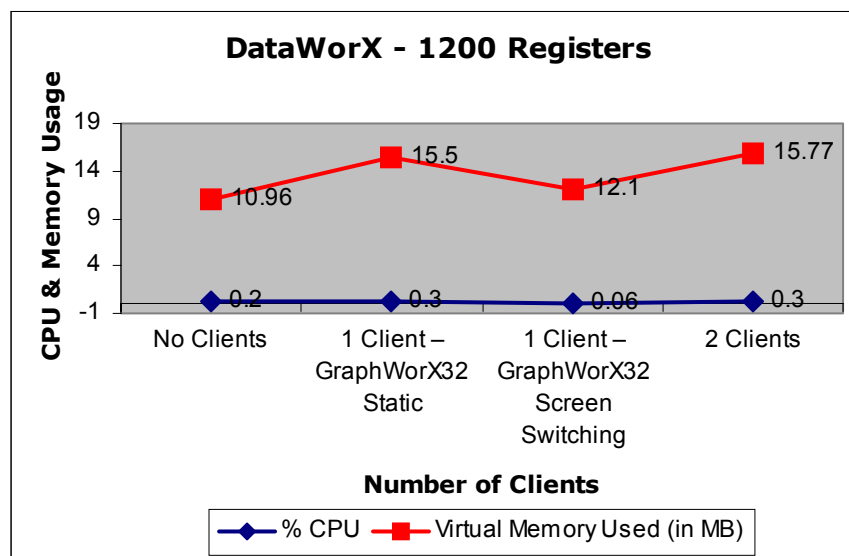


**Chart 4.15**

## 4.16 Test Case 16 DataWorX32 with 6000 Digital & Analog Registers

DataWorX32 has 6000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 6000 Digital Registers with 500 changing every 1-second, 2000 changing every 10 seconds, 5000 changing every 1 minute. The remaining 1000 are Analog Registers of which 500 change every 1 second, and 500 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) display has 50 Registers changing every 1-second, 200 Registers changing once every 10 seconds, and 250 Registers changing once every 1 minute. The other 2 displays have 500 Registers each, which are Analog Registers. Of which 250 change once a second and 250 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 1.6   | 31.24                       |
| 1 Client – GraphWorX32 Static           | 4.6   | 52.99                       |
| 1 Client – GraphWorX32 Screen Switching | 1.9   | 33.48                       |
| 2 Clients                               | 5.0   | 53.50                       |

**Table 4.16**

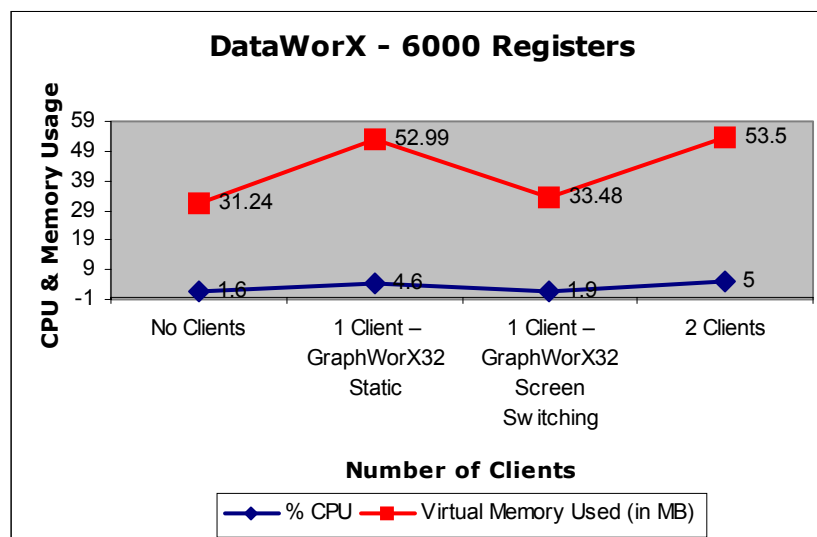


**Chart 4.16**

### 4.17 Test Case 17 DataWorX32 with 12,000 Digital & Analog Registers

DataWorX32 has 12,000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 12,000 Digital Registers with 500 changing every 1-second, 2000 changing every 10 seconds, 5000 changing every 1 minute. The remaining 2000 are Analog Registers of which 1000 change every 1 second, and 1000 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) display has 100 Registers changing every 1-second, 400 Registers changing once every 10 seconds, and 500 Registers changing once every 1 minute. The other 2 displays have 1000 Registers each, which are Analog Registers. Of which 500 change once a second and 500 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 5.7   | 56.04                       |
| 1 Client – GraphWorX32 Static           | 12.26 | 98.944                      |
| 1 Client – GraphWorX32 Screen Switching | 6.5   | 60.02                       |
| 2 Clients                               | 13.14 | 107.38                      |

**Table 4.17**

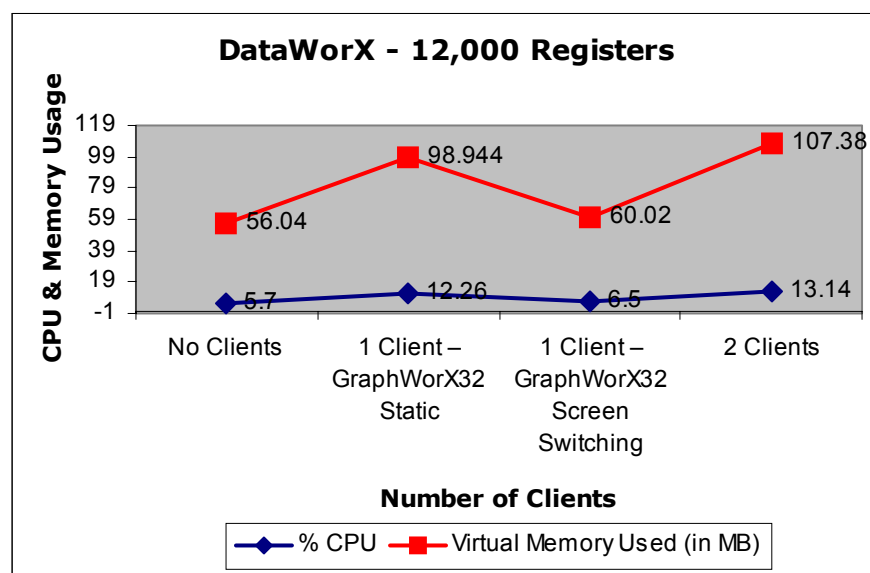


**Chart 4.17**

### 4.18 Test Case 18 DataWorX32 with 17,000 Digital & Analog Registers

DataWorX32 has 17,000 Registers in it, each changing every second. The following table shows the CPU, Memory and Virtual Memory Usage, when no clients are connected, only 1 client is connected and when 2 clients are connected.

GraphWorX32 Static Display: 12,000 Digital Registers with 500 changing every 1-second, 2000 changing every 10 seconds, 5000 changing every 1 minute. The remaining 2000 are Analog Registers of which 1000 change every 1 second, and 1000 change twice a second.

GraphWorX32 Screen Switches: Each of the 10 GraphWorX32 displays (out of 12) display has 100 Registers changing every 1-second, 400 Registers changing once every 10 seconds, and 500 Registers changing once every 1 minute. The other 2 displays have 1000 Registers each, which are Analog Registers. Of which 500 change once a second and 500 change twice every 1 second.

| Number of Clients                       | % CPU | Virtual Memory Used (in MB) |
|-----------------------------------------|-------|-----------------------------|
| No Clients                              | 4.9   | 76.16                       |
| 1 Client – GraphWorX32 Static           | 11.4  | 118.39                      |
| 1 Client – GraphWorX32 Screen Switching | 6.1   | 79.98                       |
| 2 Clients                               | 12.72 | 119.68                      |

**Table 4.18**

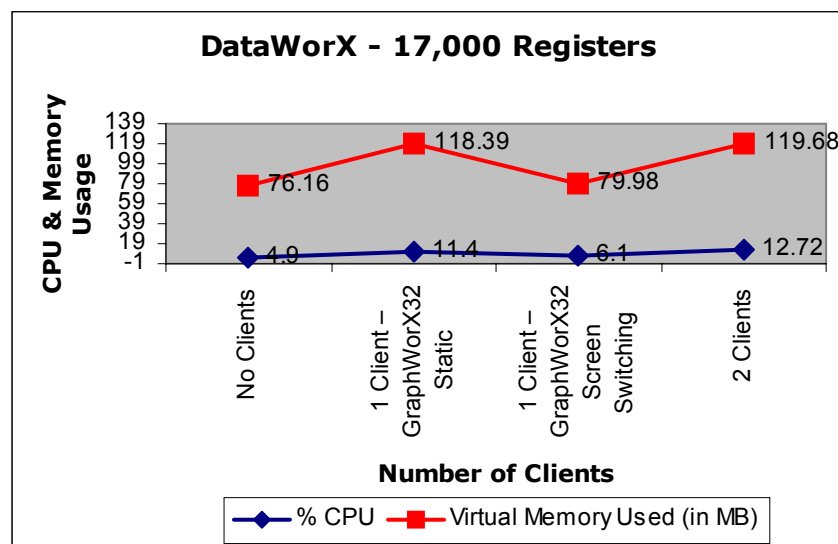


**Chart 4.18**

### 4.19 DataWorX32 Performance with 2 clients

The following chart shows the DataWorX32 performance as the numbers of tags are increased. The values are when 2 clients are requesting data from DataWorX32

| Number of Clients | % CPU | Virtual Memory Used (in MB) |
|-------------------|-------|-----------------------------|
| 1000 Registers    | 0.09  | 14.10                       |
| 1200 Registers    | 0.3   | 15.77                       |
| 5000 Registers    | 1.6   | 45.52                       |
| 6000 Registers    | 5.0   | 53.50                       |
| 10,000 Registers  | 5.3   | 84.47                       |
| 12,000 Registers  | 13.14 | 107.38                      |
| 15,000 Registers  | 5.7   | 104.12                      |
| 17,000 Registers  | 12.72 | 119.68                      |

**Table 4.19**

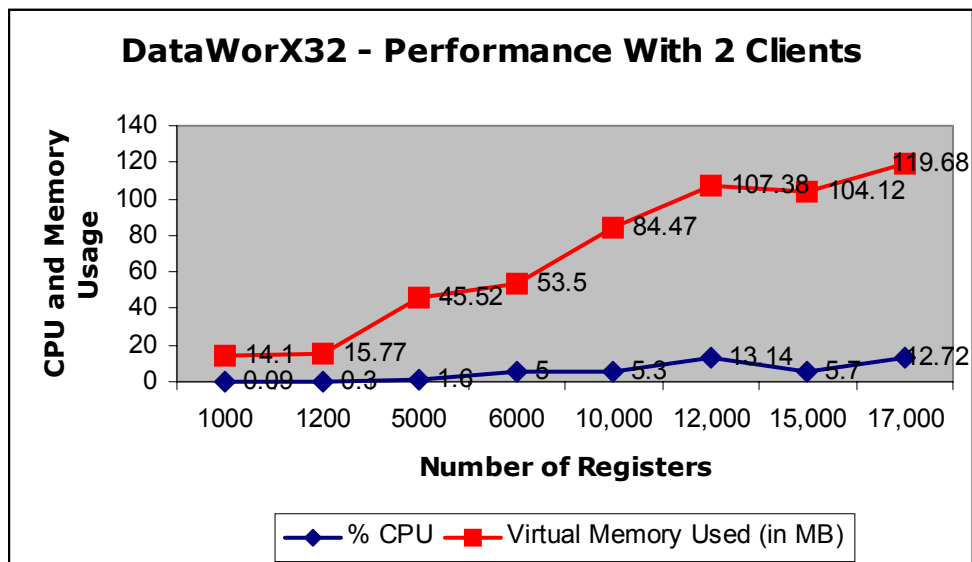
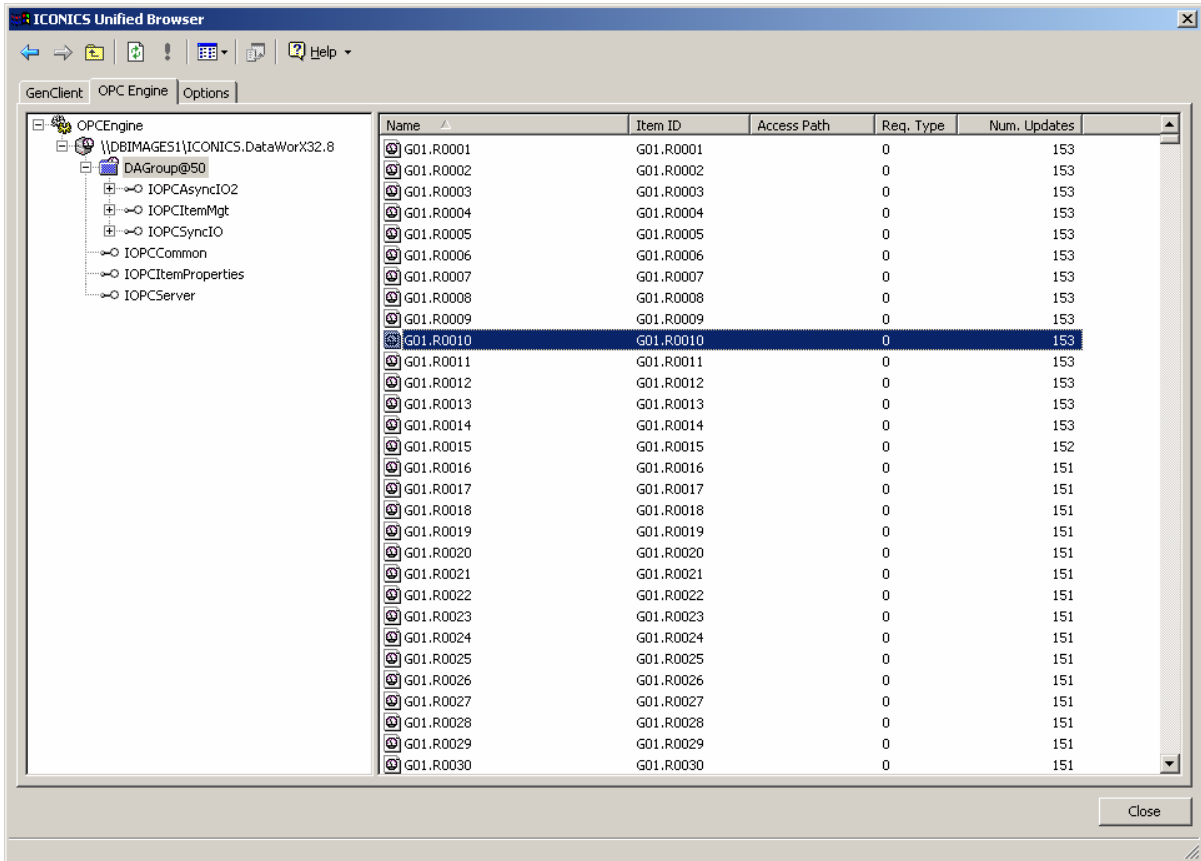


**Chart 4.19**

## 4.20 Are as many updates received as there are sent from the server?

This test can be done by checking the GenStatistics on both the server-side and client-side. What the statistics files should show is there are as many updates received as there are sent. While the files cannot be saved at exactly the same time, the figures can be slightly off. The figures below show the results of one of the tests. As can be seen the number of updates sent matches the number of updates received. (The screens show the data for Test 2 and Test 3)



| Name      | Item ID   | Access Path | Req. Type | Num. Updates |
|-----------|-----------|-------------|-----------|--------------|
| G01.R0001 | G01.R0001 |             | 0         | 153          |
| G01.R0002 | G01.R0002 |             | 0         | 153          |
| G01.R0003 | G01.R0003 |             | 0         | 153          |
| G01.R0004 | G01.R0004 |             | 0         | 153          |
| G01.R0005 | G01.R0005 |             | 0         | 153          |
| G01.R0006 | G01.R0006 |             | 0         | 153          |
| G01.R0007 | G01.R0007 |             | 0         | 153          |
| G01.R0008 | G01.R0008 |             | 0         | 153          |
| G01.R0009 | G01.R0009 |             | 0         | 153          |
| G01.R0010 | G01.R0010 |             | 0         | 153          |
| G01.R0011 | G01.R0011 |             | 0         | 153          |
| G01.R0012 | G01.R0012 |             | 0         | 153          |
| G01.R0013 | G01.R0013 |             | 0         | 153          |
| G01.R0014 | G01.R0014 |             | 0         | 153          |
| G01.R0015 | G01.R0015 |             | 0         | 152          |
| G01.R0016 | G01.R0016 |             | 0         | 151          |
| G01.R0017 | G01.R0017 |             | 0         | 151          |
| G01.R0018 | G01.R0018 |             | 0         | 151          |
| G01.R0019 | G01.R0019 |             | 0         | 151          |
| G01.R0020 | G01.R0020 |             | 0         | 151          |
| G01.R0021 | G01.R0021 |             | 0         | 151          |
| G01.R0022 | G01.R0022 |             | 0         | 151          |
| G01.R0023 | G01.R0023 |             | 0         | 151          |
| G01.R0024 | G01.R0024 |             | 0         | 151          |
| G01.R0025 | G01.R0025 |             | 0         | 151          |
| G01.R0026 | G01.R0026 |             | 0         | 151          |
| G01.R0027 | G01.R0027 |             | 0         | 151          |
| G01.R0028 | G01.R0028 |             | 0         | 151          |
| G01.R0029 | G01.R0029 |             | 0         | 151          |
| G01.R0030 | G01.R0030 |             | 0         | 151          |

Figure 4.20 a

| Name                                       | Class       | St... | Gen ... | Cli H... | Eng ... | Req. ... | Resp... | Upda... |
|--------------------------------------------|-------------|-------|---------|----------|---------|----------|---------|---------|
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0001 | OPCSecPoint | OK    | 24040   | 9450...  | 24037   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0002 | OPCSecPoint | OK    | 24042   | 1236...  | 24038   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0003 | OPCSecPoint | OK    | 24044   | 1126...  | 24039   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0004 | OPCSecPoint | OK    | 24046   | 1311...  | 24040   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0005 | OPCSecPoint | OK    | 24048   | 9208...  | 24041   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0006 | OPCSecPoint | OK    | 24050   | 1393...  | 24042   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0007 | OPCSecPoint | OK    | 24052   | 1264...  | 24043   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0008 | OPCSecPoint | OK    | 24054   | 1150...  | 24044   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0009 | OPCSecPoint | OK    | 24056   | 1342...  | 24045   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0010 | OPCSecPoint | OK    | 24058   | 9250...  | 24046   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0011 | OPCSecPoint | OK    | 24060   | 1111...  | 24047   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0012 | OPCSecPoint | OK    | 24062   | 9358...  | 24048   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0013 | OPCSecPoint | OK    | 24064   | 1176...  | 24049   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0014 | OPCSecPoint | OK    | 24066   | 1373...  | 24050   | 2        | 2       | 153     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0015 | OPCSecPoint | OK    | 24068   | 1248...  | 24051   | 2        | 2       | 152     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0016 | OPCSecPoint | OK    | 24070   | 1138...  | 24052   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0017 | OPCSecPoint | OK    | 24072   | 9399...  | 24053   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0018 | OPCSecPoint | OK    | 24074   | 1203...  | 24054   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0019 | OPCSecPoint | OK    | 24076   | 1096...  | 24055   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0020 | OPCSecPoint | OK    | 24078   | 1275...  | 24056   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0021 | OPCSecPoint | OK    | 24080   | 9157...  | 24057   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0022 | OPCSecPoint | OK    | 24082   | 9439...  | 24058   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0023 | OPCSecPoint | OK    | 24084   | 1230...  | 24059   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0024 | OPCSecPoint | OK    | 24086   | 1120...  | 24060   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0025 | OPCSecPoint | OK    | 24088   | 1304...  | 24061   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0026 | OPCSecPoint | OK    | 24090   | 9198...  | 24062   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0027 | OPCSecPoint | OK    | 24092   | 9480...  | 24063   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0028 | OPCSecPoint | OK    | 24094   | 1258...  | 24064   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0029 | OPCSecPoint | OK    | 24096   | 1145...  | 24065   | 2        | 2       | 151     |
| \\DBIMAGES1\ICONICS.DataWorX32.8\G01.R0030 | OPCSecPoint | OK    | 24098   | 1335...  | 24066   | 2        | 2       | 151     |

Figure 4.20 b

## 5 Conclusions

---

As we can see from the tables below, the performance of DataWorX32 has not changed significantly between versions 7.1 and version 8. Although DataWorX32 V8 uses the latest OPC 3.1 Server technology and runs as an NT Service.

- ✓ The reason why there is an increase in the Memory Usage is because, in V8, DataWorX32 configuration is now a Database instead of a binary file. Because of the database drivers we are seeing this increase.
- ✓ The CPU Usage appears to be the same or better in some cases for V8.0

### DataWorX32 V 7.1 Performance

| Number of Clients | % CPU | Virtual Memory Used (in MB) |
|-------------------|-------|-----------------------------|
| 1000 Registers    | 0.24  | 10.83                       |
| 1200 Registers    | 0.3   | 11.52                       |
| 5000 Registers    | 1.2   | 20.02                       |
| 6000 Registers    | 3.6   | 22.55                       |
| 10,000 Registers  | 6.7   | 36.65                       |
| 12,000 Registers  | 14.34 | 41.59                       |
| 15,000 Registers  | 7.9   | 49.36                       |
| 17,000 Registers  | 7.9   | 56.70                       |

### DataWorX32 V 8 Performance

| Number of Clients | % CPU | Virtual Memory Used (in MB) |
|-------------------|-------|-----------------------------|
| 1000 Registers    | 0.09  | 14.10                       |
| 1200 Registers    | 0.3   | 15.77                       |
| 5000 Registers    | 1.6   | 45.52                       |
| 6000 Registers    | 5.0   | 53.50                       |
| 10,000 Registers  | 5.3   | 84.47                       |
| 12,000 Registers  | 13.14 | 107.38                      |
| 15,000 Registers  | 5.7   | 104.12                      |
| 17,000 Registers  | 12.72 | 119.68                      |