

ALARMWORX32 VIEWER V8

Performance White Paper

Version 8
August 2004



OPC Automation at your fingertips

100 Foxborough Blvd., Foxborough, MA 02035
508.543.8600 • FAX 508.543.1503
www.iconics.com • E-mail info@iconics.com

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

Contents

1	About This Document	2
1.1	Scope of the Document	2
1.2	Copyright and Confidentiality	2
1.3	Revision History	2
2	Introduction	3
2.1	Overview	3
3	Test Setup	4
3.1	Hardware	4
3.1.1	3.1.1 PC1: OPC Server	4
3.1.2	3.1.2 PC2: AlarmWorX32 Server	4
3.1.3	3.1.3 PC3: Client AlarmWorX32 Viewer	4
3.2	Software	4
3.2.1	3.2.1 PC1: OPC Server	4
3.2.2	3.2.2 PC2: AlarmWorX32 Server	5
3.2.3	3.1.3 PC3: Client AlarmWorX32 Viewer	5
3.3	Network Layout	5
3.4	Testing Methodology	6
4	Test Results	7
4.1	Test Case 1 Maximum Messages without Features	7
4.2	Test Case 2 Maximum Messages with Features Enabled	8
4.3	Test Case 3 Maximum Messages in a Single burst	8
4.4	Test Case 4 Maximum Records AlarmViewer32 ActiveX can hold	8
4.5	Test Case 5 Maximum Messages without Features	8
4.6	Test Case 6 Maximum Messages with Features Enabled	9
4.7	Test Case 7 Maximum Messages in a Single burst	10
4.8	Test Case 8 Maximum Records AlarmViewer32 ActiveX can hold	11
5	Conclusions	12

1 About This Document

1.1 Scope of the Document

This document contains the performance test results for AlarmWorX32 Viewer. 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The document is © Copyright 2004, ICONICS, Inc., Foxborough, Massachusetts as an unpublished work.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 – Initial Report

Version 8.01 – RLA, August 2004

2 Introduction

2.1 Overview

The intention for this document is to summarize the performance test results for AlarmWorX32 Viewer ActiveX for Genesis V7.1 and Genesis V8.

In this test, we are measuring the performance of AlarmWorX32 Viewer ActiveX. We are measuring the performance in terms of CPU, Memory Usage and Virtual Memory Usage by AlarmWorX32 Viewer ActiveX.

We also compare the Performance between Genesis V 7.1 and Genesis V8 Beta IV.

We are in particular testing the following in the AlarmWorX32 Viewer ActiveX:

- ✓ Maximum number of messages processed per second (safely and consistently)
- ✓ Maximum number of messages processed per second with all the features enabled in Alarm Viewer.
- ✓ Maximum number of messages that can be safely received in a single instance of time (single burst)
- ✓ Maximum number of messages that can be safely maintained while allowing for the processing of the new messages

3 Test Setup

3.1 Hardware

3.1.1 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 3.1.2 PC2: AlarmWorX32 Server

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

3.1.3 3.1.3 PC3: Client AlarmWorX32 Viewer

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

3.2 Software

3.2.1 3.2.1 PC1: OPC Server

ICONICS Simulator 3.05

**For V 7.1 Test: Genesis V 7.1 and
For V8: Genesis V8 Beta IV**

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 3.2.2 PC2: AlarmWorX32 Server

For V 7.1 Test: Genesis V 7.1 and
For V8: Genesis V8 Beta IV

3.2.3 3.1.3 PC3: Client AlarmWorX32 Viewer

For V 7.1 Test: Genesis V 7.1 and
For V8: Genesis V8 Beta IV

AlarmViewer32 ActiveX – 8.0.132.2

All the communications are over **DCOM**. The client machine has GraphWorX32 with AlarmWorX32 Viewer ActiveX running on it.

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

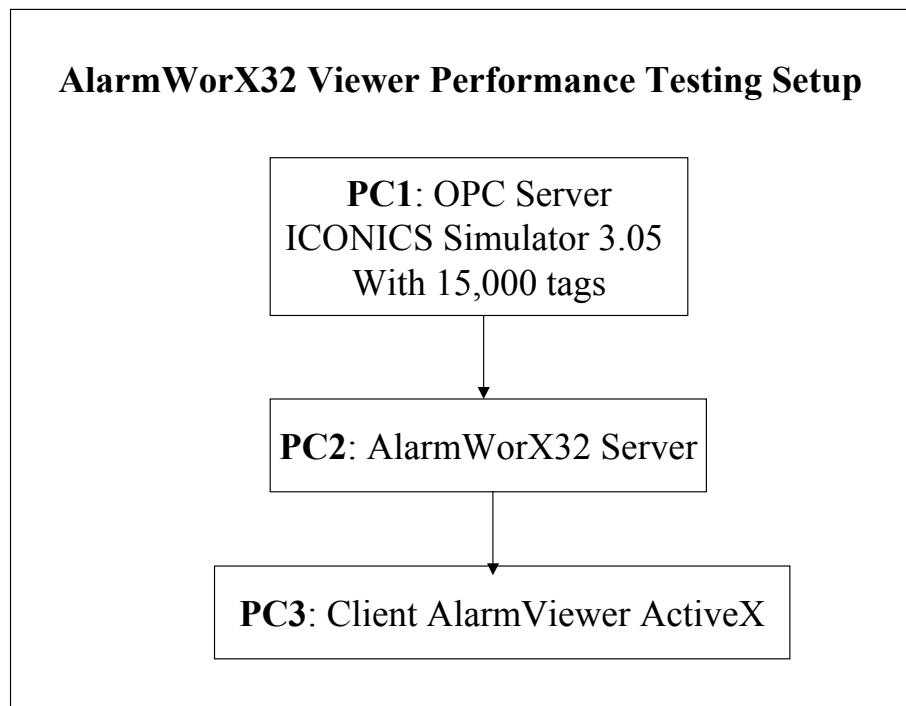


Figure 3.3

3.4 Testing Methodology

4 Tests were run for the AlarmWorX32 Viewer ActiveX, to measure the CPU and Memory Usage.

Alarm Server was running on the Server machine. GraphWorX on the client machine had AlarmWorX32 Viewer ActiveX. The display was in runtime for 15 minutes before the values of the CPU and Virtual Memory Usage were noted down. CPU Usage is got from the Performance Monitor, and Virtual Memory Used is got from the Task Manager.

Note: Test Cases 4.1 to 4.4 involve testing of Genesis V7.1
From Test Case 4.5 onwards all test cases are done in Genesis V8, Beta IV

4 Test Results

4.1 Test Case 1 Maximum Messages without Features

In the first test, AlarmWorX32 Server had 300 Alarms changing every second. On the client machine we had a GraphWorX display with an AlarmWorX32 Viewer ActiveX running on it. No features were enabled, or existing features modified for this test. It is subscribed to the remote Alarm Server. We noticed that the CPU max'ed out, after 10 minutes. So we reduced the number of alarms to 250.

With 250 Alarms/sec, we were able to process the alarms without maximizing the CPU.

Number of Alarms	% CPU	Virtual Memory Used (in MB)
250 Alarms / sec	20	10.80
300 Alarms / sec	100	NA

Table 4.1

So we can conclude that AlarmWorX32 Viewer ActiveX can process approximately **250 Alarms / Sec** in Genesis V 7.1

4.2 Test Case 2 Maximum Messages with Features Enabled

In the second test, we enabled the following features in the AlarmWorX32 Viewer ActiveX. The CPU and Memory Usage are shown in the table below:

- Show Row Header
- Show Row Number
- Set Row Height as 2
- Wrap Text
- Set the font to Bold, Italic and select a different font
- Flash when in Alarm State and change the text size, and specify new colors to flash
- 3 different sorts were enabled. First sort on Tag (Ascending), then on Time (Descending), next on Event Time (Descending)
- Enable Grid Lines
- Enable Milliseconds

The CPU and Memory Usage are shown below when AlarmViewer32 ActiveX with and without features enabled

AlarmWorX32 Viewer with 250 Alarms / sec	% CPU	Virtual Memory Used (in MB)
AlarmWorX32 Viewer	20	10.80
AlarmWorX32 Viewer with some features enabled	27	10.20

Table 4.2

4.3 Test Case 3 Maximum Messages in a Single burst

In this test, AlarmWorX32 Server had 10,000 Alarms changing every 1-minute. We put the GraphWorX32 display in runtime, to see if we can get all the alarms.

The largest number of Alarms that can be received at one time cannot exceed 8000

4.4 Test Case 4 Maximum Records AlarmViewer32 ActiveX can hold

In this test, AlarmWorX32 Server had only 5000 tags changing every 1-minute. GraphWorX32 display on the client machine had an AlarmWorX32 Viewer ActiveX running. This ActiveX for this test is subscribed to the local and remote AlarmWorX32 Servers.

It is determined that, Alarm Viewer can handle 5000 records safely and can still process new alarms.

4.5 Test Case 5 Maximum Messages without Features

In the first test, AlarmWorX32 Server had increasing number of Alarms changing every second. On the client machine we had a GraphWorX display with an AlarmWorX32 Viewer ActiveX running on it. It is subscribed to the remote AlarmWorX32 Server. The

AlarmViewer32 ActiveX has the default properties. No new properties were added or existing properties modified. The CPU and Virtual Memory Usage are shown below, as the number of Alarms in Alarm Viewer is increased.

No. Of Alarms	% CPU	Virtual Memory Usage (in MB)
1000 Alarms / sec	8.1	9.47
2000 Alarms / sec	21	10.89
3000 Alarms / sec	33	12.12
4000 Alarms / sec	46	14.03
5000 Alarms / sec	57	15.18
6000 Alarms / sec	75	16.90

Table 4.5

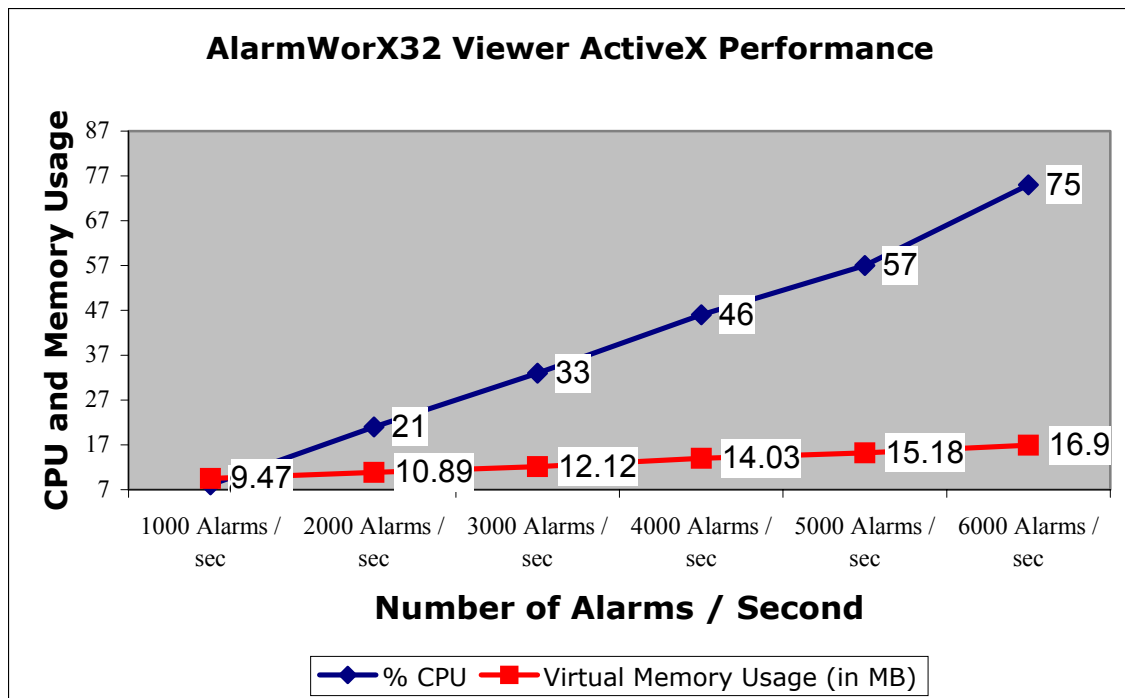


Chart 4.5

4.6 Test Case 6 Maximum Messages with Features Enabled

In the second test, we enabled the following features in the AlarmWorX32 Viewer ActiveX. The CPU and Memory Usage are shown in the table below:

- Show Row Header
- Show Row Number
- Set Row Height as 2
- Wrap Text
- Set the font to Bold, Italic and select a different font
- Flash when in Alarm State and change the text size, and specify new colors to flash
- 3 different sorts were enabled. First sort on Tag (Ascending), then on TimeinAlarm (Descending), next on Time (Descending)
- Enable Grid Lines
- Enable Milliseconds
- Show Time in Alarm
- Show Alarm Type

No. Of Alarms	% CPU	Virtual Memory Usage (in MB)
1000 Alarms / sec	11	9.85
3000 Alarms / sec	39	13.44
5000 Alarms / sec	61	20.65

Table 4.6

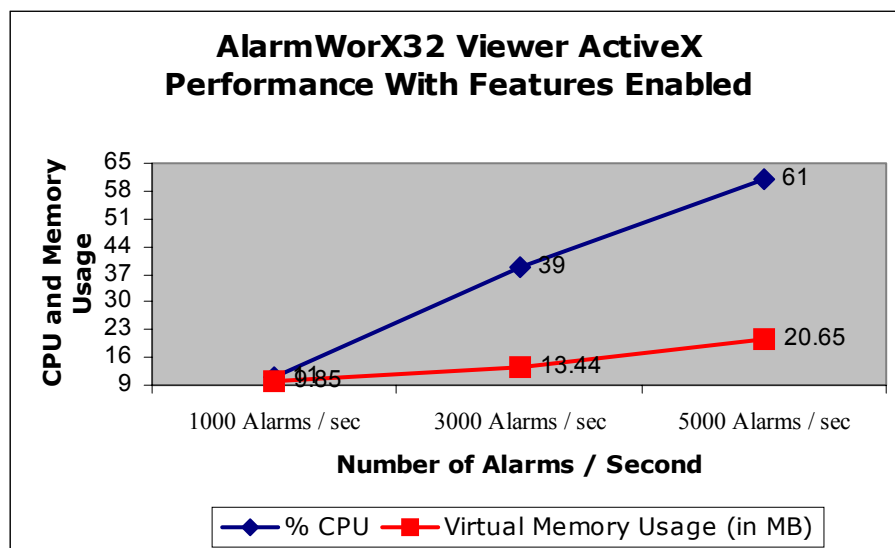


Chart 4.6

4.7 Test Case 7 Maximum Messages in a Single burst

In this test, AlarmWorX32 Server had 10,000 Alarms changing every 1-minute. We put the GraphWorX32 display in runtime, to see if we can get all the alarms.

The largest number of Alarms that can be received at one time cannot exceed 10,000.

Note: This limit is due to a set buffer in Alarm OLE.

4.8 Test Case 8 Maximum Records AlarmViewer32 ActiveX can hold

In this test, AlarmWorX32 Server had only 5000 tags changing every 1-minute. GraphWorX32 display on the client machine had an AlarmWorX32 Viewer ActiveX running. This ActiveX for this test is subscribed to the local and remote AlarmWorX32 Servers.

The maximum number of Records an AlarmWorX32 Viewer ActiveX is currently set to 32,000. This value / number can be increased by a Registry setting.

This value has been tested up to 84,000.

Note: These records are stored in the memory of the computer. So the actual number of records, Alarm Viewer can hold depends on the RAM and the CPU of the actual computer where AlarmWorX32 Viewer ActiveX is running.

5 Conclusions

After looking at the various tables and Performance Charts we can conclude that the Performance of the AlarmWorX32 Viewer ActiveX has increased by **24 times**.

The performance has increased by **2400%** in terms of

- CPU
 - Memory Usage and
 - Number of Alarms processed / sec.
- ✓ As you can see from the tables below in 7.1, the maximum number of Alarms that can be processed/sec is only 250. In V8, the number of alarms that can be processed/sec is 6000.
 - ✓ In 7.1 the CPU Usage for 250 Alarms/sec is 20%, but in V8, the CPU Usage for 1000 Alarms/sec is only 8.1%
 - ✓ In 7.1 the Memory Used for 250 Alarms is 11 MB, but in V8, we can process 2000 Alarms/sec for the same amount of Memory.

Number of Alarms	% CPU	Virtual Memory Used (in MB)
250 Alarms / sec	20	10.80
300 Alarms / sec	100	NA

Based on GENESIS32 V7.1

Number Of Alarms	% CPU	Virtual Memory Usage (in MB)
1000 Alarms / sec	8.1	9.47
2000 Alarms / sec	21	10.89
3000 Alarms / sec	33	12.12
4000 Alarms / sec	46	14.03
5000 Alarms / sec	57	15.18
6000 Alarms / sec	75	16.90

Based on GENESIS32 V8