



**Epicor ERP  
Performance Diagnostic and  
Troubleshooting Guide  
9.05.701**

## Disclaimer

This document is for informational purposes only and is subject to change without notice. This document and its contents, including the viewpoints, dates and functional content expressed herein are believed to be accurate as of its date of publication. However, Epicor Software Corporation makes no guarantee, representations or warranties with regard to the enclosed information and specifically disclaims any applicable implied warranties, such as fitness for a particular purpose, merchantability, satisfactory quality or reasonable skill and care. As each user of Epicor software is likely to be unique in their requirements in the use of such software and their business processes, users of this document are always advised to discuss the content of this document with their Epicor account manager. All information contained herein is subject to change without notice and changes to this document since printing and other important information about the software product are made or published in release notes, and you are urged to obtain the current release notes for the software product. We welcome user comments and reserve the right to revise this publication and/or make improvements or changes to the products or programs described in this publication at any time, without notice. The usage of any Epicor software shall be pursuant to an Epicor end user license agreement and the performance of any consulting services by Epicor personnel shall be pursuant to Epicor's standard services terms and conditions. Usage of the solution(s) described in this document with other Epicor software or third party products may require the purchase of licenses for such other products. Where any software is expressed to be compliant with local laws or requirements in this document, such compliance is not a warranty and is based solely on Epicor's current understanding of such laws and requirements. All laws and requirements are subject to varying interpretations as well as to change and accordingly Epicor cannot guarantee that the software will be compliant and up to date with such changes. All statements of platform and product compatibility in this document shall be considered individually in relation to the products referred to in the relevant statement, i.e., where any Epicor software is stated to be compatible with one product and also stated to be compatible with another product, it should not be interpreted that such Epicor software is compatible with both of the products running at the same time on the same platform or environment. Additionally platform or product compatibility may require the application of Epicor or third-party updates, patches and/or service packs and Epicor has no responsibility for compatibility issues which may be caused by updates, patches and/or service packs released by third parties after the date of publication of this document. Epicor® is a registered trademark and/or trademark of Epicor Software Corporation in the United States, certain other countries and/or the EU. All other trademarks mentioned are the property of their respective owners. Copyright © Epicor Software Corporation 2012. All rights reserved. No part of this publication may be reproduced in any form without the prior written consent of Epicor Software Corporation.

9.05.701

Revision: November 01, 2012 7:03 p.m.

Total pages: 56

sys.ditaval

# Contents

<b>Introduction.....</b>	<b>5</b>
Purpose of this Guide.....	5
Intended Audience.....	5
How it is Organized.....	5
<b>Epicor Performance Diagnostic Tool.....</b>	<b>7</b>
Installation.....	7
Application Server dotR File Installation.....	8
Client Diagnostics.....	9
Generate Client Logs.....	9
Analyze Client Logs.....	10
Fields and Sheets.....	11
Scenarios.....	12
Server Diagnostics.....	13
Generate Server Logs.....	13
Analyze Server Logs.....	15
Fields.....	16
Scenarios.....	18
Network Diagnostics.....	18
Enter Network Parameters.....	18
Test Network Performance.....	20
Configuration Check.....	21
Analyze Configuration.....	21
<b>Baseline Performance Tests.....</b>	<b>22</b>
Test Setup.....	22
Start the Test.....	22
Test Procedure.....	23
Activate the Trace Log.....	23
Test 1 - Baseline Network/Server Performance.....	23
Run Network and Server Test.....	23
Test 2 - Open Sales Order Entry Form.....	25
Test Form Performance Time.....	26
Verify and Fix Performance Test.....	27
Test 3 - Customer Retrieval.....	28
Run Customer Retrieval Test.....	28
Test 4 - Sales Order Line Entry.....	29
Test Sales Order Detail Line Performance.....	29
Test 5 - Purchase Order Entry.....	31
Test Purchase Order Detail Line Performance.....	31
<b>Troubleshooting List.....</b>	<b>33</b>
<b>Performance Diagnostics.....</b>	<b>39</b>

- Specific Issues.....39
  - Client Customizations.....39
  - Server Customizations.....39
  - BPM Directives.....39
  - Send Trace File to Support.....40
- Common Patterns.....40
  - Frequent User Complaints.....41
  - Everything Is Slow.....41
  - Sometimes Everything Is Slow.....41
  - Specific Program Is Slow.....41
  - Reboot Corrects Poor Performance.....42
  - Unable to Reproduce by Epicor Technical Support.....42
- Using Perfmon.....42
  - Perfmon Counter List.....43
  - Perfmon Setup.....45
  - Define Data Collectors Set Properties.....47
  - View Report.....48
- Index Enhancement Scripts.....49**
  - Verify Script: 904-500DF.....49
  - Verify Script: Epicor.9045XX.alter.index.allow\_page\_locks.sql.....50
  - Verify Script: Unique\_Index.9045XX.sql.....50
- Additional Performance Troubleshooting.....52**
  - System Troubleshooting.....52
  - Application Troubleshooting.....52

# Introduction

---

## Purpose of this Guide

---

This guide contains both the documentation for using the Epicor Performance Diagnostic Tool and the metrics you use to evaluate whether your Epicor application is performing efficiently. It has a comprehensive list of best practice recommendations as well as factors which occur over a period of time that can affect system performance.



**Important** This guide is intended for use with Epicor 9.05 or later.

## Intended Audience

---

The guide is intended for technical consultants, partners, and system administrators to help ensure the Epicor application performs as expected and to provide guidance on performance areas that should be addressed before contacting Epicor Technical Support.

Individuals who perform all or some of these tasks will benefit from reviewing the Performance Diagnostic and Troubleshooting Guide.

## How it is Organized

---

This guide explains how you can test the performance of your Epicor application.

The following are the main sections of this guide:

### **Epicor Performance Diagnostic Tool**

This first section documents how you install and use the Epicor Performance Diagnostic Tool, the key program you will use to evaluate the performance of client and server installations. This section documents how you activate the tracing logs within the Epicor application and then view them within the Performance Diagnostic Tool. It also describes the features, sheets, and fields available within the tool. Lastly, this section describes various testing methods you can leverage to evaluate performance in various areas in your system.

### **Performance Test**

This section provides detailed instructions for performing tests to measure how well your installed system performs. Epicor recommends you run these tests after the application is setup and configured to verify the Epicor application performs adequately.

You can also run these tests periodically later on to make sure the application performance has not degraded.

### **Troubleshooting Guide**

Use this section to review a comprehensive list of factors that may slow the performance of the Epicor application. These factors are related to client, application server, database, and printing activities. Consider these factors as

level one troubleshooting issues you can address before you call Epicor technical support for a more detailed investigation. Troubleshooting areas covered:

- Specific issues with one or more forms or processes
- Common patterns across the application you can observe
- General system performance issues

This section also recommends ways you can collect data Epicor technical support can use in case the issue requires more investigation. Technical support will then have better information to help you resolve the issue.



**Tip** Be sure to use this guide with the **Epicor Performance Tuning Guide - SQL** and **Epicor Performance Tuning Guide - Progress**; these guides will help you fine-tune the performance of your SQL and Progress databases. These guides are located within both the application help and the [EPICweb Customer Portal](#).

### **Perfmon Data Gathering**

Perfmon is a tool delivered with the Microsoft operating systems that can help identify system bottlenecks affecting performance. While many documents can be located on the Internet, this section focuses on Perfmon processes and counters most useful for diagnosing issues with the system performance of your Epicor application.

# Epicor Performance Diagnostic Tool

---

Use the Epicor Performance Diagnostic Tool to analyze Epicor logs. These logs help you measure performance on both the client and server installations of your Epicor application.

If you are experiencing performance issues, you should first contact either Epicor Technical Support or Epicor Consulting. If the performance issue cannot be resolved through this initial contact, the technical support representative or the consultant may send you the E9 Performance Diagnostic Tool. This tool summarizes information in both the client and server trace logs. You can manipulate this information to receive meaningful metrics that relate to the performance of your Epicor application. You can also export these results to Microsoft Excel<sup>®</sup> for additional review and analysis.

You can use this tool to measure performance against the Epicor 9.05, 9.04, and 8.03 (note that Configuration Check tab is not supported for 8.03) versions and their corresponding service packs. Leverage the Performance Diagnostic Tool to measure:

- The performance of one client versus another client on the same system within a specific location or network.
- The performance of specific business object methods on both the client and the server.
- Overall performance of the server business objects at different times in the day.
- Performance of business objects in one system against the same business objects on other systems.
- Performance of customizations and personalizations.

## Installation

---

Follow these steps to install the E9 Performance Diagnostic Tool.

Tool Requirements:

- .NET 3.5 Runtime
- Any Windows operating system



**Tip** The E9 Performance Diagnostic Tool can be installed on any desktop or laptop machine where you would normally install the Epicor client. When using the tool, you will be able to connect to application servers on other machines.

1. Unzip the tool installation files.
2. Navigate to the folder that contains the unzipped tool installation files; double-click the **Setup.exe** file.
3. The **Welcome** window appears. Click **Next**.
4. Select the installation folder you want and click Next.
5. Confirm the installation by clicking Next again.
6. Click **Close**.
7. Optionally, go to the next procedure, *Application Server dotR File Installation*.

This step is required only when you need to enable the Epicor 9 appserver configuration check utility (**Config Check** tab) on an application server machine.

You can now launch the tool from the **Start>Programs>E9 Performance Diagnostic Tool** path.

## Application Server dotR File Installation

---

Follow these steps to install the dotR file that is required to enable the Epicor 9 AppServer configuration check utility (**Config Check** tab) on the AppServer machine.

Skip this procedure in you are running Epicor 9.05.700. The dotR file is included automatically in a 9.05.700 installation.



**Important** Please check the [EPICweb Customer Portal](#) for updates to this utility, for new rules checks implemented via updated dotR files, and for updates to this guide.

1. Verify that the E9 Performance Monitoring Tool is installed.

For more information, see the topic Installation.

2. In the location where you have extracted the E9 Performance Tool files, find and extract `ConfigCheckDotR.zip` to `C:\temp` or other temporary location of your choice on the AppServer machine.

The extracted files include dotR files for all combinations of Epicor and databases.

3. Determine whether OpenEdge is 32- or 64-bit. Looking at Task Manager on the AppServer machine:

- If OpenEdge is 64-bit, you will see `_proapsv.exe` and not `_proapsv.exe*32`.
- If you see `*32` at the end it means that your OS is 64-bit but OpenEdge is 32-bit.

4. Assuming, for this example, OpenEdge 64-bit and Non-Unicode SQL-Server (32 or 64-bit SQL does not matter for this example), copy the extracted `ConfigCheck` folder to your AppServer installation:

- For a default installation, copy `C:\temp\64Bit\SQL\Server\bo\ConfigCheck` to `C:\Program Files\Epicor Software\Epicor905\Server\bo\`.

When you are done you should see the `C:\Program Files\Epicor Software\Epicor905\Server\bo\ConfigCheck` folder with the `ConfigCheck.r` file in it along with two other dotR files.

- If `C:\Program Files\Epicor905\Server` is not in your AppServer PROPATH, then pick the correct folder for your installation. Use OpenEdge Explorer to see the interactive AppServer agent's settings.
- When you are load balanced, such as behind F5 or NLB, copy the `ConfigCheck` folder into all AppServer instances.
- When you have a print server or AppServer instances for other purposes, copy the `ConfigCheck` folder into those AppServer instances.

There is no need to stop or restart the AppServer. Only the E9 diagnostic tool uses this dotR file, so it is safe to use in a production environment. If you later overwrite the dotR file with a newer version, it will be safe for production.

5. To verify the installation, perform a configuration check.

For more information, see the topic Configuration Check.



## Client Diagnostics

---

You analyze the performance of client installations through client tracing logs. This section of the guide describes how you set up these logs and analyze their results.

### Generate Client Logs

You generate the client logs by activating the trace log in the Epicor application. You then view this trace log in the Performance Diagnostic Tool.

To generate a client trace log:

1. Log onto a client installation of the Epicor application.
2. Click on **Options>Tracing Options**.  
The **Tracing Options** window displays.
3. Select the **Enable Trace Logging** check box.
4. Click **OK**.
5. Now launch some programs whose performance you want to measure. Add new records, modify records, run processes, and so on.  
The Trace Log records gathers performance data about each action you take within the Epicor application.
6. Click on **Options>Tracing Options** again.  
The **Tracing Options** window displays.
7. Click the **Write** button.  
The tracing details are copied to the log. Note the directory path within the **Current Log File** field.
8. Now launch the Performance Diagnostic Tool. Navigate: **Start>Programs>E9 Performance Diagnostic Tool**  
The **E9 Performance Diagnostic Tool** displays.
9. For the **Client Trace File Path**, click the **Browse** button to find and select client log file you just created.
10. Select the **Exclude System Calls** check box to prevent system method calls from displaying with diagnostic results.  
The Performance Diagnostic Tool will then not display values from the **GetRowsKeepIdleTime** business object method. This method runs in the background while a current form is active, but it does not indicate how long it takes other methods like Update, GetNew, and GetList to run. These method calls are generated by user activity.
11. Notice you can also change the **Log File Date Format** to select a date format used in a different locality or enter your own **Custom Format**.
12. Click on the **Generate Diagnostics** button.

You can now analyze the trace log data within the **Results** and **Summary** sheets.

## Analyze Client Logs

Use the features on the E9 Performance Diagnostic Tool to review the client results recorded on the trace log. This topic explores the features available for analyzing the client logs.

The Results grid displays the main details gathered from each trace packet, while the Summary grid displays the performance times required for each business object to run. To analyze a client trace log:

1. Click the **Results** tab.  
The Results grid displays.
2. You can filter the log results based on values that populate each column. To do this, click on the **Funnel** icon that displays on each column header:



3. When you click on an icon, a drop-down list displays. Each drop-down list contains a series of default filter options followed by specific values from the trace log:
  - **All** - The default filter, this option displays all the values for the column from the client tracing log.
  - **Custom** - Select this option to launch the **Enter filter criteria for Type** window. Use this window to define both an **Operator** and an **Operand** to filter the results. You can define as many filter conditions as you need, creating **And/Or** relationships between each custom filter.
  - **Blank** - Use this option to only display any empty rows available on the client trace log.
  - **Nonblanks** - Select this option to display all the populated rows on the selected client trace log.
  - **Column Value** - You can filter the results by selecting a specific value available in the column.
4. To help you organize the results, **Group By** the results on the grid. Click and drag a column header (for example, Object Name) to the **Drag a column header here to group by that column** area. You can drag multiple column headers to further structure the results as you need.



**Example** If you group by using the Business Object column, the grid displays all the business objects recorded on the client trace in alphabetical order.

5. You can review the following items on each business method call:
  - **Type**
  - **Object Name**
  - **Method Name**
  - **Start Date**
  - **End Date**
  - **Execution Time**
  - **File Name**
6. Click on the **Summary** tab.  
Each business object and method displays on this grid.
7. The **Trace Date/Time** and **File Name** fields display the selected trace log as well as the date and time it was run.

8. In the **Report Monitor** section, notice the **GetRowsKeepIdleTime** method. This business object runs in the background while a current form is active.



**Tip** If you shut off the System Monitor, the tracing log did not record time against the GetRowsKeepIdleTime method.

9. The **All** grid displays the summaries of the method calls recorded on the tracing log.
10. The **Calls** column indicates how many times the method sent a call to the server.
11. The execution times , in milliseconds, for each call are calculated and display in the accompanying columns:
  - **Average**
  - **Longest**
  - **Least**
  - **Total**
12. Notice you can view the results in a Microsoft Excel file. To do this, click the **Export to Excel** button.
13. Save the .xls file to directory you can access.  
You can now launch the .xls spreadsheet and review it in Microsoft Excel.
14. To remove the generated results, click the **Clear Results** button.

## Fields and Sheets

This topic documents the fields and sheets available for analyzing the client log files.

Some fields on the interface have a context menu, which is indicated by a triangle in the upper right corner of the field. To open the context menu, right-click on the field.

### Client Trace Path

Displays the location of your client trace logs. Browse to the file you wish to view within the Performance Diagnostic Tool.

### Clear Results

Click the Clear Results button to remove data from the **Results** and **Summary** sheets.

You can also continue generating data from different trace logs without clearing the data. The Summary sheet then contains results from each trace log generation.

### Custom Format

If the date format you need is not available on the **Log File Date Format** drop-down list, enter the **Language ID** for the language you use in this field. The date format linked to this language loads into the Performance Diagnostic Tool.

### Exclude System Calls

Select this check box to prevent system method calls from displaying with the diagnostic results. These system method calls do not relate to the calls generated by application data.

## Export to Excel

Click the Export to Excel button to export results to Microsoft Excel for further manipulation and analysis. The exported data is based on which grid is currently visible. For example, if the Results grid is visible, the data on this grid is exported.

## Generate Diagnostics

Click the Generate Diagnostics button to generate the log file results.

## Log File Date Format

Select the data format that matches the **Language** identifier you use when you generate client trace log files. If your format is not available from this drop-down list, enter the format you need within the **Custom Format** field.

## Results Sheet

The Results sheet contains all calls made from this client for the duration of the current trace log. This grid supports **Group By** functionality, so you can organize the results by Object and/or Method name to view execution times for all method calls.

## Summary Sheet

The Summary sheet summarizes the results for each **BusinessObject.Method**. This sheet displays the number of calls and the Average/Least/Longest duration of these method calls. The duration is measured in milliseconds. You can sort the results by selecting the column that contains the most useful information for determining the type of calls made most frequently by this client -- or which method calls consume the most time.



**Tip** The Client Diagnostics Sheet has a misleading label. The column header has text stating "Report Monitor (Ideal < 18 ms)." This text should actually state "Report Monitor (Ideal < 20-50 ms)." This value indicates the **GetRowsKeepIdleTime** value from the client should operate in less than 50 milliseconds in a well performing network.

## Scenarios

This topic contains examples of how you can measure client performance through different methods.

### **Comparison of client performance between locations and/or client systems.**

You may have specific users reporting performance issues from their systems. By running client diagnostics from multiple desktops/laptops that perform similar Epicor functions, you can determine differences in client execution times for the same method calls.

This may provide you with some things to consider with your network, especially when comparing LAN users against WAN users. CPU differences will probably not be determined, however, because the tool is capturing execution time of the method calls, not the time taken to open forms or render the data within these forms.

### **Identification of business object performance issues.**

You may suspect a business object performs poorly under certain conditions. For example, you notice that transaction entry slows down as the volume of data increases. By monitoring execution time of method calls during entry, you may be able to measure the effect.

**Identification of performance over time.**

By analyzing the execution times of similar method calls throughout the day, you should be able to identify slow periods. These slow periods may be related to network overload or server activity during those periods.

**Identification of Epicor usage from a client.**

You can analyze the log from a client to determine what activities that client engages against the Epicor application. This method also measures the frequency of calls to business object methods from that client.

**Identification of customization issues.**

Client customizations may cause unexpected or excessive calls to business object methods. For example, a poor customization could cause multiple calls to methods not made by the base application. Compare the trace logs that track the same functions with and then without client customizations; this method may identify performance issues.

## Server Diagnostics

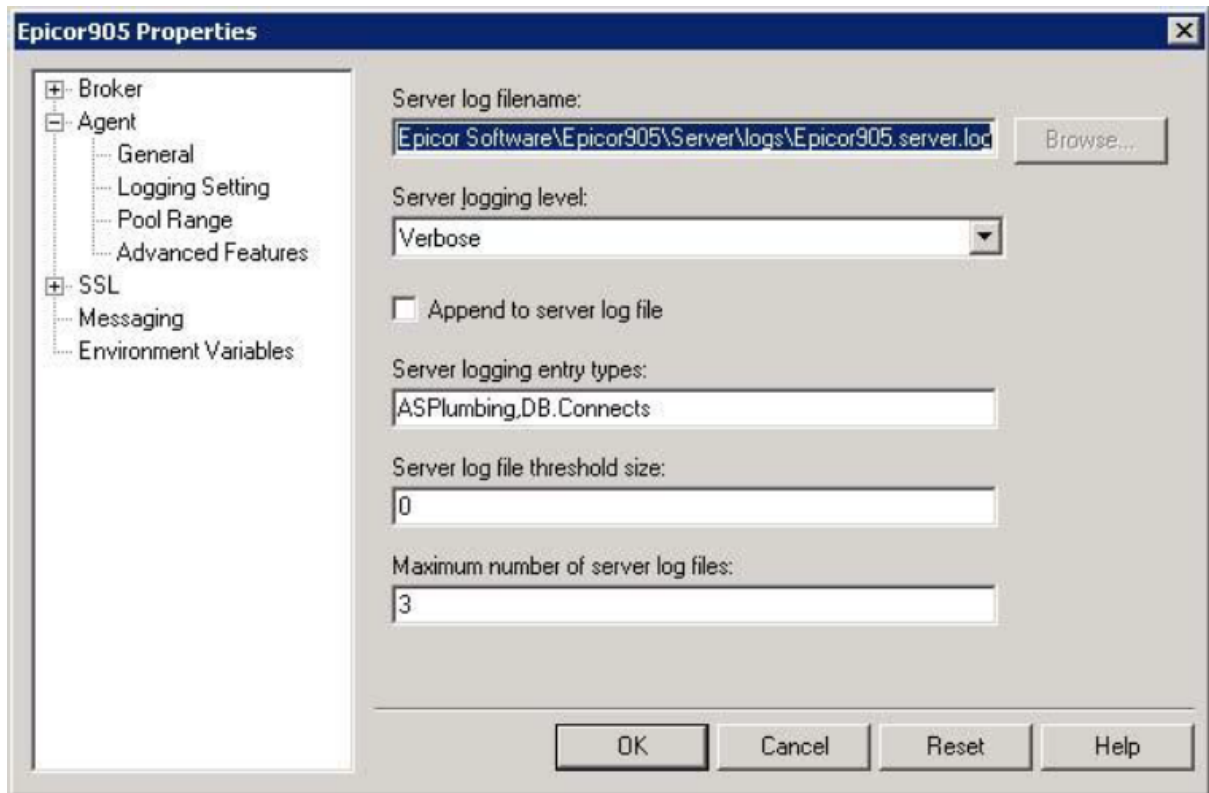
---

You analyze the performance of server installations through server tracing logs. This section of the guide describes how you set up these logs and analyze the results.

### Generate Server Logs

To generate server logs, you activate the AppServer logs you need in the Progress Explorer Tool or the OpenEdge Management Console.

1. Navigate: **Start>OpenEdge>Progress Explorer Tool**
2. Use the Progress Explorer Tool (or Management Console) to connect to the AppServer whose performance you wish to measure. Select the Server to which you want to connect.
3. Click on **Action>Connect**.
4. If you need, enter your **User Name** and **Password**. Click **OK**.  
You connect the Progress Explorer Tool to the selected server.
5. From the Tree View, expand the **AppServer** node.
6. Right-click the AppServer you wish to evaluate; from the context menu, select **Properties**.  
The Properties window displays.
7. From the Tree View, expand the **Agent** node and select the **Logging Setting** node.



8. Note the **Server log filename**: location; this field defines the directory path to the server log file.
9. Click on the **Server logging level**: drop-down list and select the **Verbose** option. This causes the log to include most server transactions.



**Tip** When you finish analyzing the server logs, Epicor recommends you set the logging level lower than Verbose. This prevents the log size from growing too large.

10. Click **OK**.

11. Navigate: **Start>Programs>E9 Performance Diagnostics Tools**

12. Click the **Add Files** button and browse to the location of the server log file.

13. Navigate to the directory location that contains the server log.



**Example** C:\Program Files\Epicor Software\Epicor905\Server\logs\Epicor905.broker.log

14. If you need, enter a value in the **Millisecond Threshold** field.  
Any calls that took longer than the millisecond value are designated as **Exceeds Threshold** in the server log results.
15. Use the **Epicor Version** drop-down list to select the application version run with the selected server log(s).  
The default version is 9.05.
16. Select the **Ignore GetRowsKeepIdleTime** check box to filter any system calls sent to the server.  
The Performance Diagnostic Tool will then not display values from the **GetRowsKeepIdleTime** business object method. This method runs in the background while a current form is active, but it does not indicate

how long it takes other methods like Update, GetNew, and GetList to run. These method calls are generated by user activity.

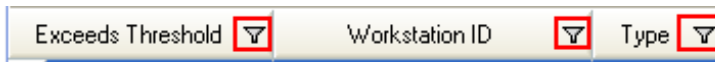
17. Now click the **Generate Diagnostics** button.

The Performance Diagnostic Tool analyzes the server log. You review this data on the **Results, Summary Analysis, Running Processes,** and **Log Errors** sheets.

## Analyze Server Logs

Use the features on the E9 Performance Diagnostic Tool to review the results recorded on the selected server log. This topic explores the features available for analyzing the server logs.

1. Click on the **Results** tab to review the server calls that were larger than the Millisecond Threshold you entered.  
The Results grid displays.
2. You can filter the log results based on values that populate each column. To do this, click on the **Funnel** icon that displays on each column header:



3. When you click on an icon, a drop-down list displays. Each drop-down list contains a series of default filter options followed by specific values from the trace log:
  - **All** - The default filter, this option displays all the values for the column from the client tracing log.
  - **Custom** - Select this option to launch the **Enter filter criteria for Type** window. Use this window to define both an **Operator** and an **Operand** to filter the results. You can define as many filter conditions as you need, creating **And/Or** relationships between each custom filter.
  - **Blank** - Use this option to only display any empty rows available on the client trace log.
  - **Nonblanks** - Select this option to display all the populated rows on the selected client trace log.
  - **Column Value** - You can filter the results by selecting a specific value available in the column.
4. To help you organize the results, **Group By** the results on the grid. Click and drag a column header (for example, Object Name) to the **Drag a column header here to group by that column** area. You can drag multiple column headers to further structure the results as you need.



**Example** If you group by using the Workstation ID column, the grid displays all the workstation entries together in alphabetical order.

5. You can review the following items on each business method call:
  - **Exceeds Threshold** - If this check box is selected, it indicates this server call took longer to execute than the **Millisecond Threshold** value you entered.
  - **Workstation ID**
  - **Type**
  - **Object Name**
  - **Method Name**
  - **Execution Time**
  - **Start DateTime**
  - **End DateTime**

- **Appserver Thread**
6. The **Summary Analysis** sheet calculates the total performance results for each business object method. Notice you can group the results by various columns on this grid.
  7. You can review several items on each business method call. Some key columns on this sheet include:
    - **Freq.** (Frequency) - This value indicates how often the business object method sent a call to the server.
    - **Total Time**
    - **Average Time**
    - **Longest**
    - **Least**
  8. The **Running Process** sheet breaks down the log by the **AppServer Threads** and **Start DateTime**.
  9. The **Log Errors** sheet displays all the errors and messages that generated in the log.
  10. Each row first indicates from which **WorkStation** the error or message originated.
  11. Review the **Errors and Messages** column to analyze the source for the error or the message.
  12. Notice you can view the results in a Microsoft Excel file. To do this, click the **Export to Excel** button.
  13. Save the .xls file to directory you can access.  
You can now launch the .xls spreadsheet and review it in Microsoft Excel.
  14. To remove the generated results, click the **Clear Results** button.

## Fields

This topic documents the fields and sheets available for analyzing the server log files.

Some fields on the interface have a context menu, which is indicated by a triangle in the upper right corner of the field. To open the context menu, right-click on the field.

### Add Files

Click the **Add Files** button to select one or more Server.log files for analysis. If the AppServer is set with a **Log File Threshold Size**, multiple Server.log files may have been generated, and these files are all available in the **Log File** folder. You can then analyze these files together within the Performance Diagnostic Tool.

You can also click the **Down Arrow** next to the Add Files button to select Server.log files from multiple server locations. You may need to do this on load balanced systems where the AppServers are located across multiple servers.

### Clear Results

Click the Clear Results button to remove the log file information from the Results and Summary sheets.

### Clear Selected

Click the Clear Selected button to remove the log file paths you have currently loaded into the Performance Diagnostic Tool.



## Epicor Version

A drop-down list currently not used by the Performance Diagnostic Tool; this feature will be available in a future release. Select an option from this drop-down list to define which Epicor version generated the current log file.

## Export to Excel

Click the Export to Excel button to export results to **Microsoft® Excel®** for further manipulation and analysis. The exported data is based on which grid is currently visible. For example, if the Results grid is visible, the data on this grid is exported.

## Generate Diagnostics

Click the Generate Diagnostics button to create the server log file data results.

## Ignore GetRowsKeepIdleTime

Click this check box to hide all **GetRowsKeepIdleTime** method call information from the server analysis results. This method call is run by the **System Monitor** to check for server related activity like uploading reports. Because this method is called frequently by every client on your system, you may not want to see this information.

## Log Errors Sheet

Any errors recorded within the Server.log file display on this sheet.

## Millisecond Threshold

Any **BusinessObject.Method** calls that are greater than the value entered in this field have their **Exceeds Threshold** check boxes automatically selected. This value indicates these calls are over the threshold value, in milliseconds, you defined. Use this value to perform **Sort By** or **Group By** actions to review longer duration method calls.

## Results Sheet

The Results sheet displays all calls received by the AppServer during the run of the server log. You can use Group By functionality to group the results by **Object** and/or **Method** to see the execution times for all method calls.

## Running Processes Sheet

This sheet displays the processes that had a **Start Time** value, but no identified **End Time** value. The Performance Diagnostic Tool assumes these processes were still running with the server log file data was generated.

## Summary Sheet

The Summary sheet summarizes the results for each BusinessObject.Method, showing the number of calls and the Total/Average/Least/Longest duration of those calls. These duration values are measured in milliseconds. You can sort the results by column to determine the types of calls made most frequently against the server, or which calls consume the most time.

## Scenarios

This topic contains examples of how you can measure server performance through different methods.

### Identification of server performance compared to other servers.

Use this method to identify performance differences for similar business object method calls on different servers. You do this by comparing the elapsed times for the method calls on one server against another server. These time differences may be related to CPU, database, customizations, operating systems, hardware configurations, and other differences.

### Analysis of activity by workstation.

If you group by on the **Workstation** column, you can identify the method calls made by specific workstations. If a user reports slow performance, you can identify what this user was doing at the time and whether the elapsed times for method calls run using expected elapsed times.

### Analysis of system performance over time.

If you group by using the **Object Name** and **Method Name** columns, you can observe the performance of specific method calls throughout the day. You may be able to discover slow periods by observing overall differences in execution time patterns for those methods.

### Analysis of most common activities and performance.

If you sort by the **Frequency** column on the **Summary** sheet, you can view the most frequent method calls and their execution times. Sort by the **Total Time** column to see what activities consume the most time on the server.

### Analysis of server verses network elapsed time for methods.

On the **Results** sheet, if you group by the **WorkStation** column, you can see the execution times of specific method calls on that workstation. You should also be able to identify the same calls from the Client Diagnostics for that workstation and compare the differences. For example, if you see an elapsed time of 800 milliseconds for a server call and you see 1500 milliseconds for that same call on a client, you can conclude that about 700 milliseconds was spent transferring data over your network. If you do not like this network performance, you can analyze network traffic using a network analysis tool.

### Identification of customization issues.

You can analyze the elapsed time it takes to call specific methods compared to other server installations. This may indicate issues with customizations, business process method (BPM) method/data directives, or ABL code customizations.

## Network Diagnostics

---

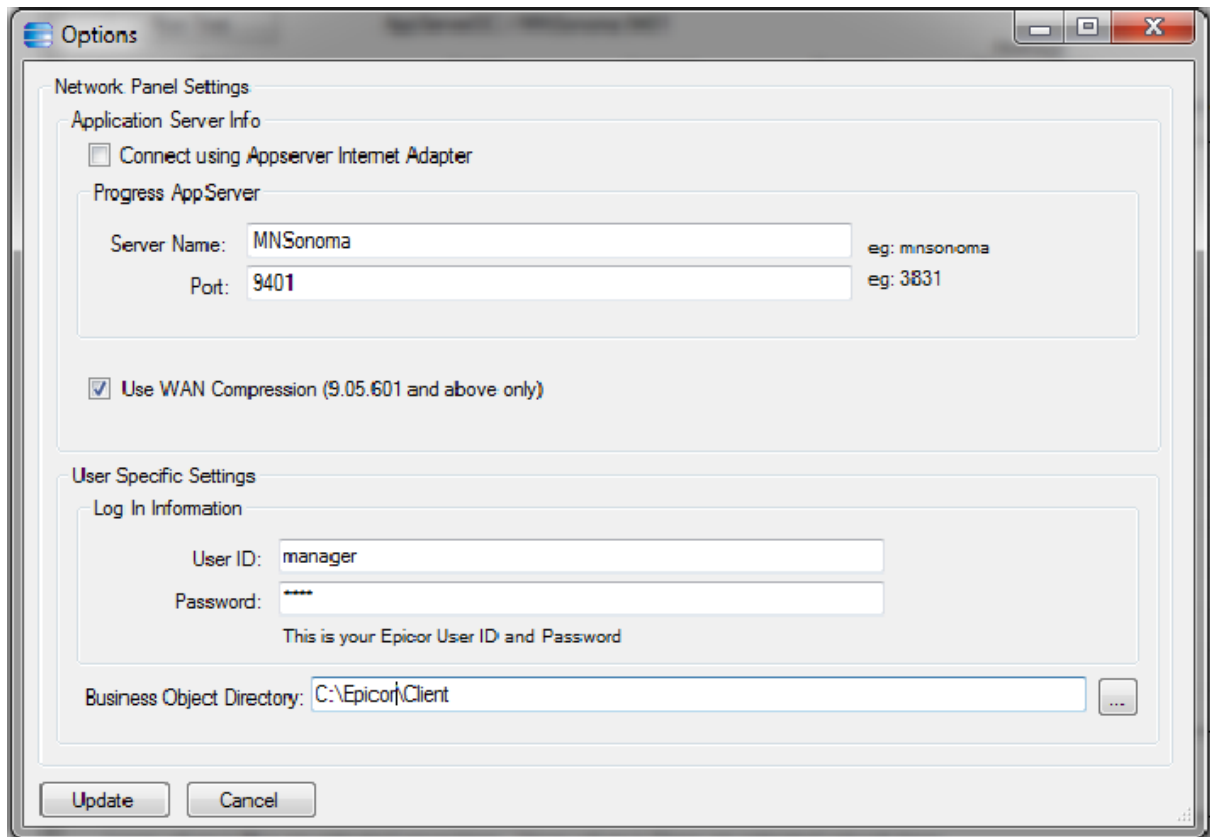
You can use the E9 Performance Diagnostic Tool to verify the baseline network and server performance are running at optimal levels.

### Enter Network Parameters

1. Navigate: **Start>Programs>E9 Performance Diagnostic Tool**  
The E9 Performance Diagnostic Tool window displays.

2. Click **Tools>Options**.

The **Options** window appears.



3. If you use the AppServer Internet Adapter to connect to the application server, select the **Connect using Appserver Internet Adapter** check box.

The Progress AppServer prompt messages are replaced by a prompt from the connection URL.

4. Enter the **Server Name** value you need. This value is the name of your application server.

5. Next, define the **Port** number you use to connect to the application server.

6. If you run the network over a WAN, select the **Use WAN Compression** check box. This option is only available for Epicor 9.05.601 or later versions.



**Important** If you are on a Local Area Network (LAN) or you are running the tool on the application server, you do not need to select the WAN Compression check box.

7. Enter the **User ID** and **Password** for an Epicor ERP user who can connect to the application server.

8. Now define the **Business Object Directory**. This value is the fully qualified path to the directory that contains the Epicor ERP client assemblies.

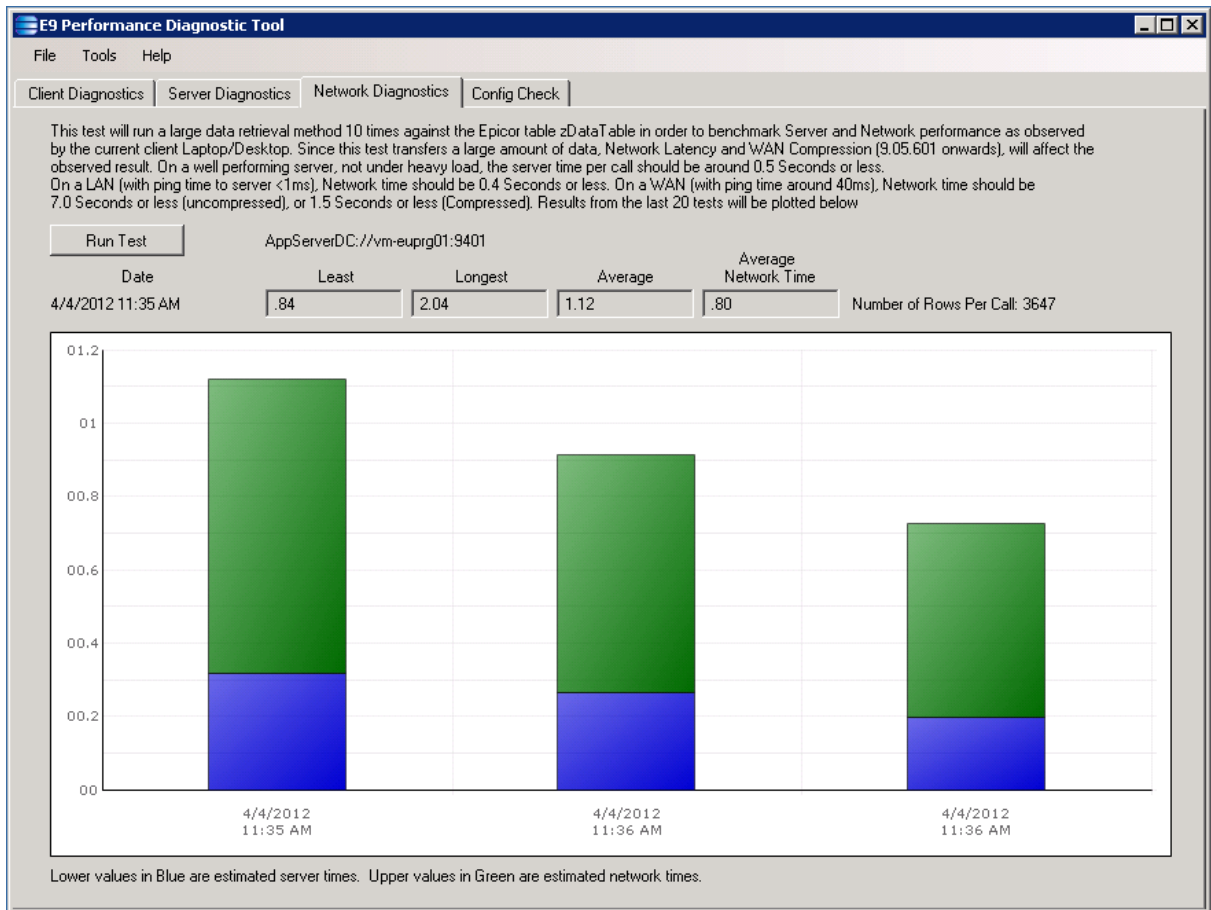
9. Click the **Update** button.  
Your option settings are saved.

## Test Network Performance

Run this test to verify that the baseline network and server performance is running at optimal levels.

1. Navigate to the **Network Diagnostics** sheet.
2. Click the **Run Test** button three times.

Three bar graphs appear on this sheet. You can click on each bar graph to view the specific details of each network test.



Expected Results:

- Server Time (blue/lower bar) < 0.5 Seconds
- Network Time (green/upper bar) < 0.4 Seconds over a LAN
- Network Time (green/upper bar) < 7 Seconds without compression or < 1.5 seconds with compression over a WAN

## Configuration Check

---

The E9 Performance Diagnostic Tool contains a utility to check the configuration of the Epicor 9 AppServer.

Use the Config Check sheet to evaluate the AppServer configuration. After the Performance Diagnostic Tool analyzes the configuration, this feature displays recommended actions you can take to fix various issues.

### Analyze Configuration

The Config Check sheet contains tools for analyzing your AppServer configuration settings.

1. Navigate to the **Config Check** sheet.



**Tip** Notice this sheet is linked to the AppServer and Port you defined on the Options sheet. For more information, review the Enter Network Parameters topic.

2. Click the **Check Configuration** button.
3. The **Config Check Summary** sheet displays a list of rules run against which the AppServer was checked.
4. The **Results** column displays the generated evaluation of each rule against your AppServer configuration. Available results include FAIL, WARNING, EXISTS, and PASS.
5. Depending on the results, different instructional text displays in the **Action Required** column.
  - **EXISTS** – Notifies you that various items, like customizations and BPM directives, are active in the current system. These items should be evaluated for performance.
  - **FAIL** – The AppServer configuration did not meet the rule requirements. The ActionRequired column displays a recommended action you can do.
  - **PASS** – The configuration met or surpassed these rule requirements. No further action is needed.
  - **WARNING** – Alerts you that potential performance issues may occur. Review these items to see if further changes are needed.
6. The **Config Check Details** sheet displays the various rule keys run to evaluate each configuration rule. Expand one of the rules to see the specific calls.
7. Notice you can view the results in a Microsoft Excel file. To do this, click the **Export to Excel** button.
8. Save the .xls file to directory you can access.  
You can now launch the .xls spreadsheet and review it in Microsoft Excel.
9. To remove the generated results, click the **Clear Results** button.

# Baseline Performance Tests

---

This section outlines a series of tests you can run on your Epicor application to verify its performance against established metrics. These metrics were defined using a test machine.

Configuration of the test machine:

- Windows Server 2008R2 64Bit
- CPU Dual Core Intel xeon 5160 @ 3GHz
- 20 GB RAM
- Epicor SQL database installed on separate database server with 1G bit/sec Ethernet connection
- Epicor version 9.05.601

## Test Setup

---

Be sure to run all tests on the delivered base Training database. You should not have any active Business Process Management (BPM) directives or customizations running against the business objects you will test.

If you receive significant variations in the results, this may indicate that BPM directives or customizations are degrading the application performance.

These tests are intended primarily to measure server and database performance. To eliminate variations caused by the network, run all of these all tests over a LAN (Local Area Network) with a ping time (32 bytes) to the Application Server < 1 ms. If you test with a higher network latency, it may result in some measurable difference in the observed results.

## Start the Test

Do the following steps to begin the test.

1. Install E9 Performance Diagnostic Tool on your server machine as described in the previous section.
2. Log into your Epicor application using both the Training database and the manager account. For the User ID, enter manager; for the Password, enter manager.
3. From the Main Menu tree view, navigate to the Epicor USA company.
4. Now open the Main Plant node.
5. Close the **System Monitor**. If the System Monitor is running in the Task Tray, then right click on this icon and select **Exit**. This eliminates **GetRowsKeepIdle** time calls to the server.

## Test Procedure

---

The following test procedure provides you with a repeatable path that uses standard data delivered within the Training database.

The results from the tests are captured in the Client Trace log. You can then analyze these results using the E9 Performance Diagnostic Tool.

### Activate the Trace Log

Use these steps to activate the trace log.

1. From the Main Menu, select **Options>Tracing Options**.  
The **Tracing Options Form** appears.
2. Select the **Enable Trace Logging** check box.  
This activates the trace log.
3. Notice the **Clear Log** button. At certain points during the following tests, you will be asked to click this button.  
Clicking this button removes results from the trace log.
4. Notice the **Write** button. At certain points during the following tests, you will be asked to click this button.  
Clicking this button causes business object calls to be recorded in the database.

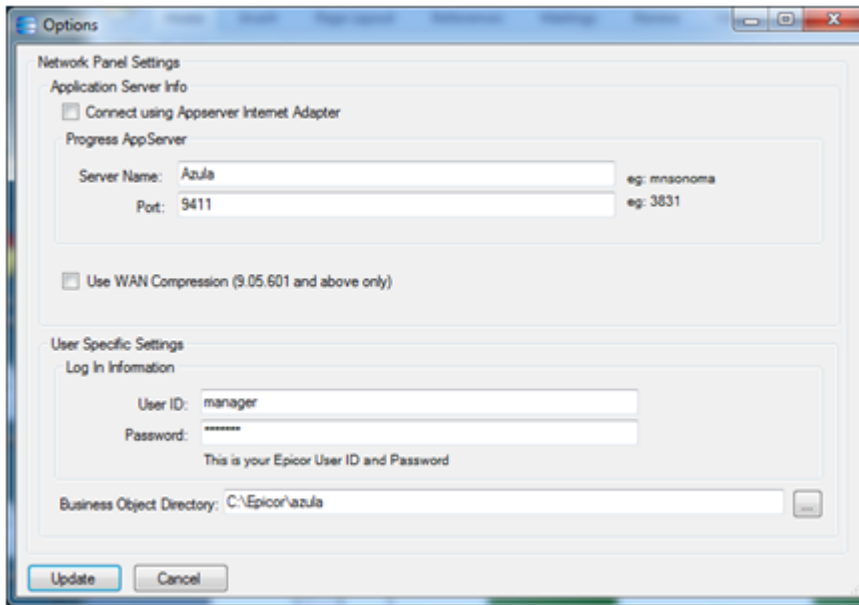
## Test 1 - Baseline Network/Server Performance

---

Run this test using the E9 Performance Diagnostic Tool to verify that baseline network and server performance is adequate to support the remaining tests.

### Run Network and Server Test

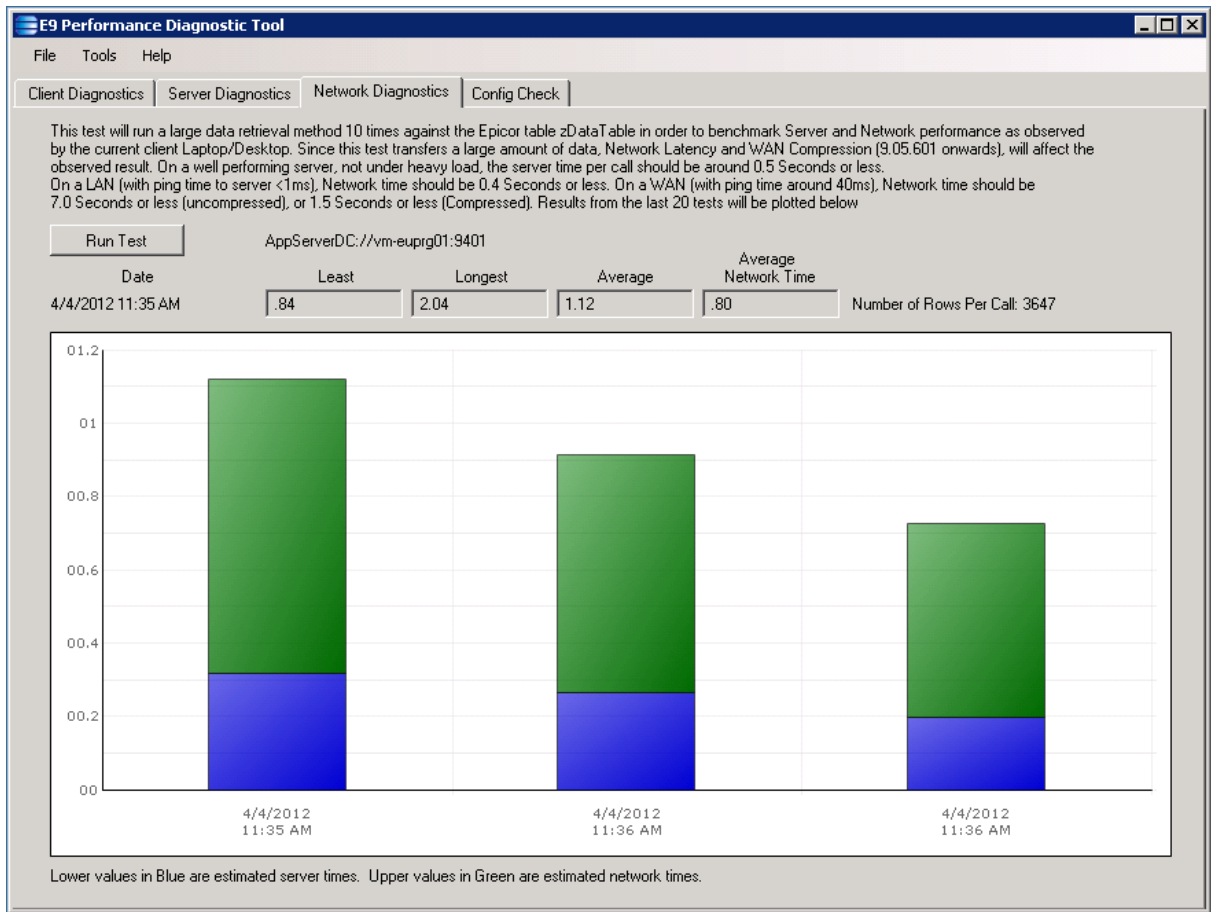
1. Launch the **E9 Performance Diagnostic Tool**.
2. Click **Tools>Options**.  
The Options window displays.
3. Complete the Server Connection details as shown in the below illustration:



**Important** Because you are either on a Local Area Network (LAN) or on a server, you do not need to select the WAN Compression check box.

4. Click **Update**.
5. Close the **Options** window.
6. Navigate to the **Network Diagnostics** sheet.
7. Click the **Run Test** button three times.





**Tip** If you do not see the Network Diagnostics sheet, you need to download a more recent version of the E9 Performance Diagnostic Tool.

Expected Results:

- Server Time (blue/lower bar) < 0.5 Seconds
- Network Time (green/upper bar) < 0.4 Seconds

## Test 2 - Open Sales Order Entry Form

This test measures the observed time it takes for you to open a typical large form. This test will use Sales Order Entry.

You will clear the **Client Cache** to measure how long it takes for the form to load initially after a new installation or service pack/patch upgrade. You will then run the test twice to measure the time it takes the form to load using both uncached memory and then cached memory.

After you complete the tests, you will write the client trace log file and use the E9 Performance Diagnostic Tool to analyze the results.

## Test Form Performance Time

1. Navigate to the Sales Order Entry form: **Sales Management > Order Management > General Operations > Order Entry**
2. Launch Sales Order Entry.
3. Within the form, navigate to **Tools > Options**.
4. Verify the **Memory Cache** checkbox is not selected (clear).
5. Close Sales Order Entry.
6. From the Main Menu, select **Options > Clear Client Cache** to clear the memory cache.
7. Clear the **Trace Log** as described in the previous Test Setup section.
8. Using a stopwatch to record the elapsed time, launch Sales Order Entry again.
9. Finish stopwatch recording the moment the form displays on your screen and the cursor flashes in the Sales Order field.
10. Record this elapsed time for the **First time form download and open** value.
11. Close Sales Order Entry.
12. Once again, use a stopwatch to record how long it takes for the Sales Order Entry form to display and the cursor to appear. Launch Sales Order Entry.
13. Record this elapsed time for the **Second time form open** value.
14. Write to the **Trace Log** as described in the previous Test Setup section. Note the Log File name; this value uses the TraceDataxxx.log file format.
15. Launch the E9 Performance Diagnostic Tool.
16. Navigate to the Client Diagnostics sheet.
17. Browse to the **Client File Trace Path** to locate the client trace file you wrote as described previously.
18. Click the **Generate Diagnostics** button to capture the performance results.

Expected Results:

- First time form download and open: < **8 seconds**
- Second time form open: < **6 seconds**

Breakdown by business object from the Summary sheet:

Type	Object Name	Method Name	Calls	Average	Longest	Least	Total
BO	GenXData	GetList	1	94	94	94	94
BO	GenXData	GetRows	1	31	31	31	31
BO	CreditCardProc	GetList	2	52	31	31	156
BO	SalesOrder	GetPlantConfCtrlInfo	2	62	62	62	124
BO	SalesOrder	CheckICPO	2	93	109	78	187
BO	SalesOrder	GetCalcQtyPref	2	54	62	47	109
BO	SalesOrder	SetCalcQtysPref	2	62	62	62	124
BO	Company	GetByID	2	444	452	437	889

If you see more object calls than above and have much slower performance, it may be because the Sales Order Entry form properties are not cached. If this is the case, you may see additional **Lib** type calls as shown below:

Type	Object Name	Method Name	Calls	Average	Longest	Least	Total
Lib	ClientCache	GetClassInformation	15	386	1466	31	5802
Lib	ClassAttribute	GetAttributes	1	16	16	16	16
BO	GenXData	GetList	1	31	31	31	31
BO	GenXData	GetRows	1	31	31	31	31
Lib	BOReader	GetList(UOM)	1	31	31	31	31
Lib	BOReader	GetRows(Company)	2	413	421	406	827
Lib	BOReader	GetRows(Currency)	1	187	187	187	187
BO	CreditCardProc	GetList	2	41	62	31	124
BO	SalesOrder	GetPlantConfCtrlInfo	2	54	62	47	109
BO	SalesOrder	CheckICPO	2	54	62	47	109
BO	SalesOrder	GetCalcQtyPref	2	62	62	62	124
BO	SalesOrder	SetCalcQtysPref	2	62	62	62	124
BO	Company	GetByID	2	444	468	421	889

## Verify and Fix Performance Test

Use the following steps to verify and fix issues with the form performance test.

1. Locate the **xxx.mfgsys** configuration settings file used by the client installation. This file is typically located in the **Config** folder under the client direction.
2. Using a text editor or similar program, open the xxx.mfgsys file.
3. Locate the **<appsettings>** section
4. Review the MaxBOMRU and MaxClsAttrMRU settings.  
Default values:
  - <MaxBOMRU value=" 100 " />
  - <MaxClsAttrMRU value=" 20 " />
5. Locate the cached **Business Object** and **Class Attribute** .xml files.  
Directory paths on a Windows 7 client:
  - C:\ProgramData\Epicor\<server-port>9.05.601\EPIC03\BOsecMRUList\BOMRUList\_<username>.xml
  - C:\ProgramData\Epicor\<server-port> \9.05.601\EPIC03\ClsAttrMRUList\ClsAttrMRUList\_<username.xml>
6. If the SalesOrder business object does not appear in either of these xml files, it implies that other business objects are used more frequently on this client and are not cached. To fix this situation, do the following:
  - a. Log off the Epicor application.
  - b. Delete the two xml files.

- c. Log back onto the Epicor application.
- d. Launch Sales Order Entry.
- e. Log off the Epicor application.
- f. Review the two xml files. Among the other business objects, you should see a reference to the Sales Order business object.
- g. Repeat the Sales Order Entry form test.

## Test 3 - Customer Retrieval

---

This test measures database retrieval time by selecting and paging through customers.

You will start the tracing log to capture the time taken to select 10 customers and page through them within the Customer Maintenance form. After you complete this test, you will write to the client trace log file and use the E9 Performance Diagnostic tool to analyze results.

### Run Customer Retrieval Test

1. Navigate to the Customer Maintenance form: **Sales Management > Order Management > Setup > Customer**
2. Launch **Customer Maintenance**.
3. Click the **Customer** button to launch the search form.
4. Clear the **Trace Log** as described in the previous Test Setup section.
5. Return to the search window and click the **Search** button to retrieve customer records.
6. Click the **Select All** button.
7. Click **OK** to pull in all of the customer records into the Customer Maintenance form.
8. Using the **Navigation** toolbar, click the **Right Arrow** button nine times to display the next nine customers.
9. Write to the **Trace Log** as described in the previous Test Setup section. Note the Log File name; this value uses the TraceDataxxx.log file format.
10. Launch the E9 Performance Diagnostic Tool.
11. Navigate to the Client Diagnostics sheet.
12. Browse to the **Client File Trace Path** to locate the client trace file you wrote as described previously.
13. Click the **Generate Diagnostics** button to capture the performance results.

Expected Results:

- Observed time to move between customers: < **1 second**

Breakdown by business object from the Summary sheet:

Type	Object Name	Method Name	Calls	Average	Longest	Least	Total
BO	Customer	GetBillDayList	11	41	47	31	453
BO	SalesTer	GetList	1	47	47	47	47
BO	EpiSearch	DoNamedSearch	1	78	78	78	78
BO	QuickSearch	GetBaseDefault	1	16	16	16	16
BO	Customer	GetListCustom	1	109	109	109	109
BO	Customer	GetByCustID	10	335	359	312	3355
BO	CustCnt	GetRows	10	56	109	31	560
BO	CustCnt	GetCustCntGlobalFields	2	31	31	31	62

## Test 4 - Sales Order Line Entry

---

This test measures database update performance. During this test, you will create a sales order that contains twenty detail lines.

You will enter a sales order Header and then start the client tracing to capture the time it takes to enter the 20 lines. After you complete the test, you will write the client trace log file and use the E9 Performance Diagnostic Tool to analyze the results.

### Test Sales Order Detail Line Performance

1. Navigate to Sales Order Entry: **Sales Management > Order Management > General Operations > Order Entry**
2. Launch Sales Order Entry.
3. To create a new sales order, click the **New** button.
4. For the **Sold To ID** value, enter DALTON.
5. Clear the **Ready to Process** check box (if it is selected).
6. Click **Save** to record the order.
7. Verify that the first five column headers on the **Lines > List** sheet match the column headers in the table below.

If not then rearrange the columns to match this

Line	Part	Rev	Desc	Order Quantity
0	AV-TA			12
0	CA10			10
0	CA12			12
0	CA20			20
0	DCD-300-OR			20
0	DCD-400-KB			100
0	DCD-600-CDL			600
0	DSS-1030			10
0	G1000			1000
0	MRP100-SK			100
0	PC1			420
0	PC7			480
0	SD-TA			79
0	4600-1			200
0	OOP1			10
0	OOP2			20
0	OOM1			100
0	OOM3			30
0	OLM1			10
0	OLM2			20

8. Create a spreadsheet that contains these twenty sales order detail lines (or copy lines from this electronic document).
9. Clear the **Trace Log** as described in the previous Test Setup section.
10. Copy the twenty detail line from your spreadsheet into your clipboard. Do not select the column headers.
11. Right click above the column headers in the Lines>List sheet; select **Paste Insert** from the context menu.
12. Wait while the twenty order detail lines are loaded into the Sales Order Entry form.
13. Write to the **Trace Log** as described in the previous Test Setup section. Note the Log File name; this value uses the TraceDataxxx.log file format.
14. Launch the E9 Performance Diagnostic Tool.
15. Navigate to the Client Diagnostics sheet.
16. Browse to the **Client File Trace Path** to locate the client trace file you wrote as described previously.
17. Click the **Generate Diagnostics** button to capture the performance results.

Expected Results:

- Total observed time for 20 lines: **72 Seconds** (measured from Results tab as difference between start time of first method and end time of last method)

Breakdown by business object from the Summary sheet:

Type	Object Name	Method Name	Calls	Average	Longest	Least	Total
BO	SalesOrder	GetNewOrderDtl	20	330	515	281	6615
BO	SalesOrder	ChangePartNumMaster	20	489	577	452	9782
BO	SalesOrder	ChangeSellingQuantity	20	382	499	343	7659
BO	SalesOrder	ChangeDiscountPercent	20	325	421	296	6506
BO	SalesOrder	MasterUpdate	20	1755	2106	1357	35101
BO	SalesOrder	GetJobProd	20	81	109	78	1623

## Test 5 - Purchase Order Entry

This test measures database update performance. During this test, you will create a purchase order that contains twenty detail lines.

You will enter a purchase order Header and then start the client tracing to capture the time it takes to enter the 20 detail lines. After you complete the test, you will write the client trace log file and use the E9 Performance Diagnostic Tool to analyze the results.

### Test Purchase Order Detail Line Performance

1. Navigate to Purchase Order Entry: **Material Management > Purchase Management > General Operations > Purchase Order Entry**
2. Launch Purchase Order Entry.
3. To create a new purchase order, click the **New** button.
4. For the **Supplier ID** value, enter ABCM.
5. Click **Save** to record the order.
6. Verify that the first seven column headers on the **Inventory** sheet match the column headers in the table below.

If not then rearrange the columns to match this

Line	Type	Part	Rev	Description	Our Quantity	UOM
		516X075B			10	EA
		1032x075			20	EA
		1032X100			20	EA
		38X075B			10	EA
		38X100B			10	EA
		38X125B			10	EA
		38X200			20	EA
		516X075B			30	EA
		516X100B			30	EA
		516X150B			30	EA
		516X200B			30	EA
		84005-042			5	EA
		84005-053			5	EA
		84005-060			5	EA
		84005-116			5	EA
		84005-610	A		5	EA
		516FW			30	EA
		516LN			30	EA
		10w30Lube			10	EA
		8mm-1.25H Nut			10	EA

7. Create a spreadsheet that contains these twenty purchase order detail lines.

8. Clear the **Trace Log** as described in the previous Test Setup section.
9. Copy the twenty detail line from your spreadsheet into your clipboard. Do not select the column headers.
10. Right click above the column headers in the Inventory sheet; select **Paste Insert** from the context menu.
11. Wait while the twenty purchase order detail lines are loaded into the Purchase Order Entry form.
12. Write to the **Trace Log** as described in the previous Test Setup section. Note the Log File name; this value uses the TraceDataxxx.log file format.
13. Launch the E9 Performance Diagnostic Tool.
14. Navigate to the Client Diagnostics sheet.
15. Browse to the **Client File Trace Path** to locate the client trace file you wrote as described previously.
16. Click the **Generate Diagnostics** button to capture the performance results.

Expected Results:

- Total observed time for 20 lines: **51 Seconds** (measured from Results tab as difference between start time of first method and end time of last method)

Breakdown by business object from the Summary sheet:

Type	Object Name	Method Name	Calls	Average	Longest	Least	Total
BO	PO	GetNewPODetail	20	218	250	203	4378
BO	Part	PartIsSalesKit	20	46	62	31	920
BO	PO	PartStatusValidationMessages	20	53	62	47	1060
BO	PO	ChangeDetailPartNum	20	394	421	374	7893
BO	PO	ChangeDetailCalcOurQty	20	222	468	187	4448
BO	PO	CheckLOC	20	260	296	234	5212
BO	PO	Update	20	1135	1201	1092	22713
Lib	BOReader	GetCodeDescList	3	21	31	16	63
Lib	ClientCache	GetClassInformation	1	172	172	172	172
Lib	BOReader	GetList(ProjPhaseSearch)	3	36	47	31	109
Lib	BOReader	GetRows(COA)	1	94	94	94	94



## Troubleshooting List

This section provides a comprehensive list of possible causes for poor performance at the client, application server, database and printing tier areas of your Epicor application.

Application Tier	Probable Cause of performance issue	Best practice / Solution recommendation
Client	Not enough RAM	Plan to have at least 2 Gig of RAM on your workstation. If your workstation is used for other applications which can consume lot of RAM and CPU then increase the RAM and your workstation CPU capacity.
Client	Lots of processes running on your desktop	<p>If you have lots and lots of processes running on your desktop then the performance of Epicor ERP client will be affected like any other application's would. Review your list of processes from the task manager and stop or uninstall any unwanted processes / services. The lower the number of processes running on your desktop the better response time you will get. Of course we are not suggesting you stop any application which is crucial to run your business only stop unnecessary ones.</p> <p>If possible don't keep too many windows open on your desktop. This is not to suggest that you cannot multi-task, just a suggestion to close any unwanted application that could be eating up the precious CPU, RAM, Network and disk I/O resources.</p>
Client	Spike in CPU utilization	<p>Analyze what process (es) are consuming CPU on your desktop using task manager. The "CPU Time" column of task manager will tell you that information. If you sort it in descending order of "CPU" column it will tell you what process is currently utilizing the CPU.</p> <p>One example - If you have a virus scanner which kicks in a particular day or a particular time of the day then it will impact the performance of Epicor client. How bad it will be depends on how powerful your client box is.</p>
Client	Network speed	<p>If the network speed between the Client and the appserver box is not fast then your client response will be slow.</p> <p>If the ping time between your Client and the appserver box is above 50-100ms then you will see slower performance compared to somebody who has better ping time. Lower # the better.</p>
Client	Processor type	Budget permitting, you should try to be on a better client in terms of CPU, RAM and disk I/O. Newer Windows 7 box will perform better than older box.
Client	Terminal server	If you have too many users allocated per terminal server machine then your performance will suffer.
Client	VM Client	If your desktop is a virtual machine then performance will suffer if the host machine is overcommitted in terms of CPU and / or RAM and / or Network bandwidth.

Application Tier	Probable Cause of performance issue	Best practice / Solution recommendation
Client	DirectX	Install the latest version of DirectX.
Client	Disk I/O	If your disk is under heavy usage then it can have an impact on performance.
Client	Disk I/O	Occasional defragmentation of disk helps with overall performance and will certainly help Epicor client as well.
Client	Virus / Malware	Needless to say Virus and / or Malware infection affects overall system performance which includes Epicor ERP.
Appserver	Configuration	OpenEdge - You should have latest official OpenEdge version and hotfix on all the appservers that constitute your Epicor ERP installation
Appserver	Configuration	<p>Schema holder Set schema holder to Read-Only. Instructions <b>Support Page 8782MPS.</b></p> <p>If you are using 3<sup>rd</sup> party tools which don't work with read-only schema holder than you can launch a separate schemaholder for the 3<sup>rd</sup> party application but still continue to use Read-only schemaholder for the Epicor ERP appserver.</p>
Appserver	Configuration	<p>Appserver pool - Make sure the pool range is set correctly for the number of users accessing the appserver. You should have enough appserver agents ready for handling Epicor ERP user requests.</p> <p>The appserver can launch new appserver agents as required but we should not rely on that feature too much because it takes a while for the appserver agent to spawn and be ready for use.</p> <p>Don't set the minimum server setting too low either. This setting will cause the appserver agent to trim if not in use.</p> <p>As a rule of thumb 1 appserver agent can handle 8-10 interactive user requests. This assumes that the reports are running a separate report appserver. If not, 1 appserver can support 5 interactive and report user.</p> <p>Also don't go below 10/10/50 (Initial / Minimum / Maximum servers) for production instance.</p>
Appserver	Configuration	MfgSys.pf file (name can differ on installations) has a setting called -T. This points to the appserver temp directory. Check to see there is enough space available in this directory. Delete any old files.
Appserver	Configuration	<p>Mfgsys.pf file (name can differ from one installation to another).</p> <p>In that file:</p> <p>Make sure you have -ttmarshal 5 defined.</p> <p>Make sure you have -Bt parameter with 40960 value.</p> <p>Make sure you have -tmpbsize 8</p> <p>Make sure you have -lkwtmo 180</p> <p>Make sure you have -debugalert</p>

Application Tier	Probable Cause of performance issue	Best practice / Solution recommendation
Appserver and Database server	Configuration	Follow the <b>Hardware Sizing Guide</b> recommendation and split the appserver and SQL Server. This guide is available for download from the <a href="#">EPICweb Customer Portal</a> .
Appserver and Database server	Sizing	Download the latest Hardware Sizing Guide from the <a href="#">EPICweb Customer Portal</a> and match your hardware with the recommendation for Epicor 9.05. If it doesn't match or exceed the recommendation you *maybe* potentially under sized.
Appserver	Sizing - RAM	Have enough RAM available for all the appserver agents that will start (_proapsv.exe). Budget half Gig per agent.  If you have too many environments running on the appserver then you will run into performance problems. The test / training / pilot Epicor ERP environment should be on a separate box. That test box doesn't need to be very powerful but it should be on a separate box.
Appserver	Service Connect load	If you have service connect load serviced by the production appserver then you can run into performance problem. The best practice recommendation would be to have a separate appserver for service connect client and if the service connect load is high then put another machine for service connect appserver.
Appserver and Database server	Too many processes / services running on the appserver	Stop any unnecessary services and processes from the appserver production box.  Appserver and Database server should not have IIS and other search utility run on the box.
Appserver and Database server	Too many requests are being processed on the appserver	It is expected behavior that during the busy time of the day when the appserver / database is under heavy use that performance will be slower than when not many people are using the system.  Some large running processes like MRP can slow down the system. Now you can use MRP during the busy time of the day that's technically possible but if you can schedule it during the less busy time of the day then it will help all users.  Backup and large download of files can also slow down the processing of the appserver.
Appserver and Database Server	Virus scanner	Tune your virus scanner to ignore appserver temporary files and Epicor ERP software files. Have it not scan the entire box during peak production hours.
Appserver and Database Server	Backup / download / Windows update	These activities should be controlled and if possible avoid them during peak production hours.
Appserver and Database server	Windows OS	64-bit Windows 2008 R2 recommended.
Appserver and Database server	Windows 2008 uses aggressive memory caching.	If you are on Windows 2008 and you are concerned about memory utilization of Epicor ERP appserver and / or the whole server in general, it's recommended you upgrade to Windows 2008 R2. Of course our product will run fine on Windows 2008. Then why

Application Tier	Probable Cause of performance issue	Best practice / Solution recommendation
		upgrade? The memory usage information is easy to read in Windows 2008 R2.
Appserver and Database server	Network speed between appserver and database	<p>There should be absolutely no network latency between the appserver and SQL Server box. The connection should be direct and should not be passing thru too many network switches. If there is a network problem your performance will suffer. This is very important.</p> <p>Troubleshooting networking issues is beyond the scope of this guide. Customers have asked their network engineers to use Wireshark or other software to look for network issues including malformed and dropped network packets.</p> <p>If you can put a direct cable between the appserver and SQL Server box then it's even better. We understand it may not be possible always but you should atleast try to see if it can be done.</p>
Appserver and Database server	Network card	You need two 1 Gig network cards per appserver box. One card dedicated to getting data from SQL server and another card which services the client request.
Appserver and Database server	BIOS Setting	<p>For your production server, change the BIOS power settings to Maximum Performance. This setting is available on newer processors approx. servers purchased 2008 and beyond.</p> <p>In BIOS look for Power profile and choose "Maximum Performance".</p>
Appserver and Database server	BIOS Setting - disable C-States	Change the BIOS setting of "Minimum Processor Idle Power State" to No C-states.
Database server	Configuration	SQL Server - You should have the latest SQL Server fixes installed for the supported version of SQL Server. For example, for your SQL Server 2008 R2 should have all the released patches from Microsoft.
Database server	Too many databases hosted on the database server.	<p>If you have too many databases running on the server then you can potentially run into performance problems. The test / training / pilot Epicor ERP environment should be on a separate box. That test box doesn't need to be very powerful but it should be on a separate box.</p> <p>If somebody is running MRP or other large process on the test / training environment which is hosted on the production environment then production system's performance will suffer.</p>
Database server	Index enhancement script	<p>Run SQL Unique Index Script. The script changes the Progress_recid index of tables to be non-clustered.</p> <p><b>Support Page 10377MPS.</b></p>
Database server	Disallow page locking	<p>Run SQL Disable Allow Page Lock script. For specific tables, this script disables ALLOW_PAGE_LOCKS, sets it to off.</p> <p><b>Support Page 10908MPS.</b></p>

Application Tier	Probable Cause of performance issue	Best practice / Solution recommendation
Database server	Text catalog option	Check the Full Text Catalog option and make sure it's set to: Do not track changes mode. <b>Support Page 9043MPS.</b>
Appserver and Database server	DEP setting	Changed the DEP setting to "Turn on DEP for essential Windows programs and services only".
Database server	Disk I/O	The SQL Server files should be split across multiple spindles. Spreading the I/O load across multiple disks will improve performance. Example, Put Epicor database MDF file on a separate RAID 10 array. Put Epicor database LDF file a separate RAID 10 array. Create around 8 tempdb files and put them on a separate disk array.
Database Server	Disk I/O and RAID 5	RAID 5 is not recommended. This is important. RAID 10 is great for performance of SQL files.
Database Server	MAXDOP	In SQL Server properties change Maximum Degree of Parallelism to 1 or 2. Zero is default and is not recommended for Epicor ERP which is an OLTP application.
Database Server	snapshot_isolation_state	We recommend snapshot_isolation_state to be OFF. You can verify it using sys.databases DMV on SQL Server instance.
Database Server	read_committed_snapshot	We recommend read_committed_snapshot to be false / OFF. You can verify it using sys.databases DMV on SQL Server instance.
Database Server	In-house application and 3 <sup>rd</sup> party access via ODBC	If you are using Epicor ERP database to query Epicor ERP tables then you have to be careful you are not locking / blocking Epicor ERP transactions. The safest recommended way is to read Epicor ERP tables via Views. Create the Views with (nolock) hint and when you write select statements against the views you will not block Epicor ERP transactions. The other benefit of views is you can only include required columns in the views and this way you can improve the scalability of SQL server and improve performance of ODBC queries.
Printing	Configuration	On Epicor client, change the System Monitor Configuration Duration to 30,000 milliseconds on the client machines. The default value is 15000 milliseconds.
Printing	Configuration	Administrator of Epicor ERP will have to change Processing Delay column value from 10 seconds to 3 seconds. This can be changed using System Agent Maintenance screen.

Application Tier	Probable Cause of performance issue	Best practice / Solution recommendation
Printing	Document size	Large document will print slower than smaller documents. Use the report's selection criteria to control the amount of processing you want to do and hence the size of the document.
Printing	Spike in printing requests	If too many jobs are printed at once then you will see a queuing of requests on the appserver. If this is normal part of your application usage pattern then create a separate appserver machine for reporting. On the reporting appserver start at least 20 or 30 appserver agents in the appserver pool. 30/30/50 will do it if your hardware supports it. Processing delay should be 1 second.
Printing	Network speed	<p>The reports are generated on the appserver and then client pulls the XML or text data and renders the report on the client.</p> <p>In your System Agent Maintenance Screen you have 3 fields which end with field name "Directory" e.g. Client Directory. From the client machine if it takes a long time to retrieve files from these directories then the printing will be slow. Here again small files will download faster than the big ones and hence the data file size (report output size) will determine the time taken to print.</p> <p>If your end-users are accessing the XML data over the slow WAN then if budget permits and business needs justifies the expense, consider WAN accelerator.</p>
Printing	Custom report	<p>If your report has customization then verify the customization.</p> <p>Too many sub-reports in Crystal report will slow down the report.</p> <p>If you are retrieving too many records for printing that will also consume time.</p>
Printing	Custom report	<p>If you have written custom reports that go against the database directly then make sure you are accessing the data using efficient SQL queries.</p> <p>Use no-lock hints in your SQL statements. If that's not possible then access the records via views. You can create views with no-lock hint.</p>

# Performance Diagnostics

---

## Specific Issues

---

If you apply the best practice recommendations in the previous Troubleshooting List and you are still experiencing performance issues, you will need to explore more deeply to discover the cause. To do this, you must narrow down the possibilities to discover the specific issue causing the problem.

Begin by first identifying which entry program, maintain program, report, or process is slow. You next need to determine whether this poor performance is running on a base installation, or if the program is running against any customizations or BPM directives.

In each case described in this section, use the E9 Performance Diagnostic Tool to analyze the Client trace and verbose Server logs to identify differences in performance.

### Client Customizations

Use this technique to test the performance of a client customization against the base program version.

Make sure the user account currently logged into the application has Customization privileges. Then turn on customization mode from the Main Menu by either clicking **Options>Developer Mode** or click the **Developer Mode** button on the toolbar.

Navigate to the customized program and launch it. When the **Select Customization** window appears, select the **Base Only** check box. Click OK and time how long it takes to launch the program. Close the program and launch it again, selecting the customization. Time how long it takes the customized version of the program to launch.

### Server Customizations

Use this technique to test the performance of a server customization against the base program version.

Review server customizations in the **Epicor905/Custom** or **Epicor905/CSG** folders. If customizations exist, then temporarily rename these folders so they will not be found by the ProPath configuration. Now time the launch times of the programs within the Epicor application to see if their performance is improved.



**Important** Be sure you do not rename folders on a live system, however, or you may create other issues with the Epicor application.

### BPM Directives

Use this technique to test the performance of a BPM directive that may be negatively affecting the program.

Identify whether any BPM Method or Data directives are interacting with the process. If you do discover a BPM method is running against the process, you can disable it through either the **Method Directives** or **Data Directives** windows. You can also temporarily rename the **Epicor905/BPMExec/TrainingBPM** folder (or a similar folder) so that these are not found by the ProPath configuration.



**Important** Be sure you do not rename folders on a live system or you may create other issues with the Epicor application.

## Send Trace File to Support

If you are working with Epicor Technical Support to improve performance, follow these steps to send the trace log to a support representative.

1. During a period of peak activity on your network, launch the Epicor application on the client workstation.
2. From the Main Menu, click **Options>Tracing Option**.  
The **Tracing Options Form** appears.
3. Select the **Enable Trace Logging** check box.
4. Select the **Track Changes Only** check box.
5. Click the **Clear Log** button.
6. Now launch the program that has poor performance, following the interface movement pattern and data entry pattern that demonstrates the issue.



**Tip** Typically you should have the end user who reported the issue follow the pattern that causes the poor performance. The data being entered is also very important. The performance issue can be triggered by many factors -- the data the end user enters, the options the end user selects, navigation path the end user follows, and so on.

7. Return to the Tracing Options Form.
8. Click the **Write** button.
9. Save the log file to a location on your C:\ drive.
10. Now perform a second test during a period of slow activity on your network (after hours or on a weekend). Launch the Epicor application from the **AppServer** machine.  
The reason you launch the Epicor application during this time and environment is so you can eliminate load condition and network traffic between the client and the AppServer. Even if you know the network is not the cause of the poor performance, you must record these performance times so Epicor Technical Support can compare the performance under these different conditions.
11. Repeat steps 2-9.
12. Email the file to Epicor Technical Support.

## Common Patterns

---

This section describes some common patterns for poor performance, and the causes for these patterns.

If you are experiencing any of these common patterns, be sure to use the E9 Performance Diagnostic Tool as needed to help analyze the situation.



## Frequent User Complaints

Some users frequently complain about a performance problem. They experience poor performance regularly, but other users at this organization do not experience the same issues.

You can turn on the trace log for the whole day on the end-user client machine that has the performance problem. You will most likely need to do this for several days to capture the performance data. Be sure to also clear the trace log file each day; if you do not, the trace log will grow too large and cause performance problems. If your trace log file expands too quickly, try to save the trace log every two hours and start a new trace log.

When the end user notifies you that the Epicor application had poor performance during the current day, you will have the trace data you need. Save the trace log for that day. If you needed to save multiple trace logs for the same day, you must review all of the trace logs.

If you are working with Epicor Technical Support, send the trace log (or multiple trace logs) to the technical support representative helping you with the issue.

## Everything Is Slow

Everything is slow all of the time.

The most likely cause behind this pattern is the hardware sizing for the production environment is not correct. Use the **Hardware Sizing and Configuration Guide** to determine the recommended hardware; this guide evaluates data transaction traffic, number of users, and the system software to determine the recommended hardware.

Epicor also suggests you review the **Troubleshooting List** in the Performance Diagnostic and Troubleshooting Guide to verify the environment is configured correctly.

## Sometimes Everything Is Slow

Sometimes everything slows down, but other times the performance is fine.

This performance pattern could be caused by a number of different reasons. Most likely, the slow performance occurs because the Epicor application was experience heavy data loading at that time. The other likely possibility is that either a third party ODBC connection locked a record or the Epicor application has some other locking issue. Review the Troubleshooting List to identify and correct locking issues.

Additional information is also available from Epicor Technical Support:

- **Support Page 5861MPS** - This documentation describes how to set up AppServer logging to both verbose and -debugalert.
- **Support Page 8035MPS** - This documentation describes how to set up and run the SQL locks logger utility.

## Specific Program Is Slow

A specific program is slow.

The causes for this pattern are described in the previous Frequent User Complaints section. You will need to turn on the trace log and have the end user follow the interface or data entry pattern again to recreate the poor performance. You then have a trace log or multiple trace logs to review.

## Reboot Corrects Poor Performance

Everything slows down, but a system reboot restores good performance.

This common pattern can be caused by several issues:

- Typically several large reports or processes were generating at the same time, harming performance. These reports/processes stop when you exit and reboot the Epicor application. Determine which reports and/or processes were running and evaluate whether your network has sufficient resources to handle simultaneously running these report/processes.
- Was a long generating process, such as MRP, running at the time your users experienced poor performance? If so, try to schedule the process at a time when less demand is being placed against the network.
- Was a long generating process, such as MRP, started on a different database (Test, Training, Pilot) being still run from the same server as your live database? If this occurring, consider running this process on a different server.
- Perhaps one or multiple business activity queries (BAQs) were running simultaneously, causing the slow performance. Even one BAQ can cause a problem if both the SQL Server and the AppServer are run from the same machine. If this is the case, try to run these BAQs at a time when less demand is being placed against the network. Evaluate whether your network has sufficient resources to handle running these BAQs.
- Does your system have a memory leak? The client and server machines can then rapidly run out of memory, causing the slowdown.
- Are your records getting locked? Record locking can be caused either by the Epicor application or by third party applications. Verify that you have the **-lkwtmo 180** value defined within your **mfgsys.pf** file.

## Unable to Reproduce by Epicor Technical Support

Performance is slow on your server, but the Epicor test systems cannot reproduce the issue.

If Epicor Technical Support is unable to recreate the performance issue, consider the following:

- Evaluate the differences in hardware. Your hardware sizing needs to be evaluated by you and an Epicor Consultant or Epicor Technical Support.
- Different data items are being used, and so the test is not comparing similar conditions.
- The poor performance occurred when the system was experiencing heavy data load, so the performance issues were caused by this load.
- Review the Troubleshooting List to verify all of these items are correctly configured.

## Using Perfmon

---

This section describes how you can do more advanced troubleshooting using Perfmon. This diagnostic tool is delivered with the Windows Server operating system.

Leverage this information to learn how to collect some Windows performance counters to analyze the performance problem from a CPU, Disk I/O, RAM and Network point of view.

## Perfmon Counter List

The counters described in the following list are useful for diagnosing performance issues with your Epicor application.

After you collect the data for a few days using Perfmon, analyze the bottlenecks. You should be able to discover from where the performance pressure is coming from -- CPU, Memory, Disk, or Processor.



**Tip** To learn how to set up Perform, review the next Perform Setup section.

Object	Counter	Description	Comment / Values
Processor	% Processor Time	% Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then subtracting that value from 100%. (Each processor has an idle thread that consumes cycles when no other threads are ready to run). This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. It should be noted that the accounting calculation of whether the processor is idle is performed at an internal sampling interval of the system clock (10ms). On today's fast processors, % Processor Time can therefore underestimate the processor utilization as the processor may be spending a lot of time servicing threads between the system clock sampling interval. Workload based timer applications are one example of applications which are more likely to be measured inaccurately as timers are signaled just after the sample is taken.	Roughly below 75% is good to have. Beyond that users start seeing impact of load.
Processor	% Privileged Time	% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.	Roughly below 5-10%
Memory	Available Mbytes	Available MBytes is the amount of physical memory, in Megabytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free and zero page lists.	Greater than 10 to 50 MB

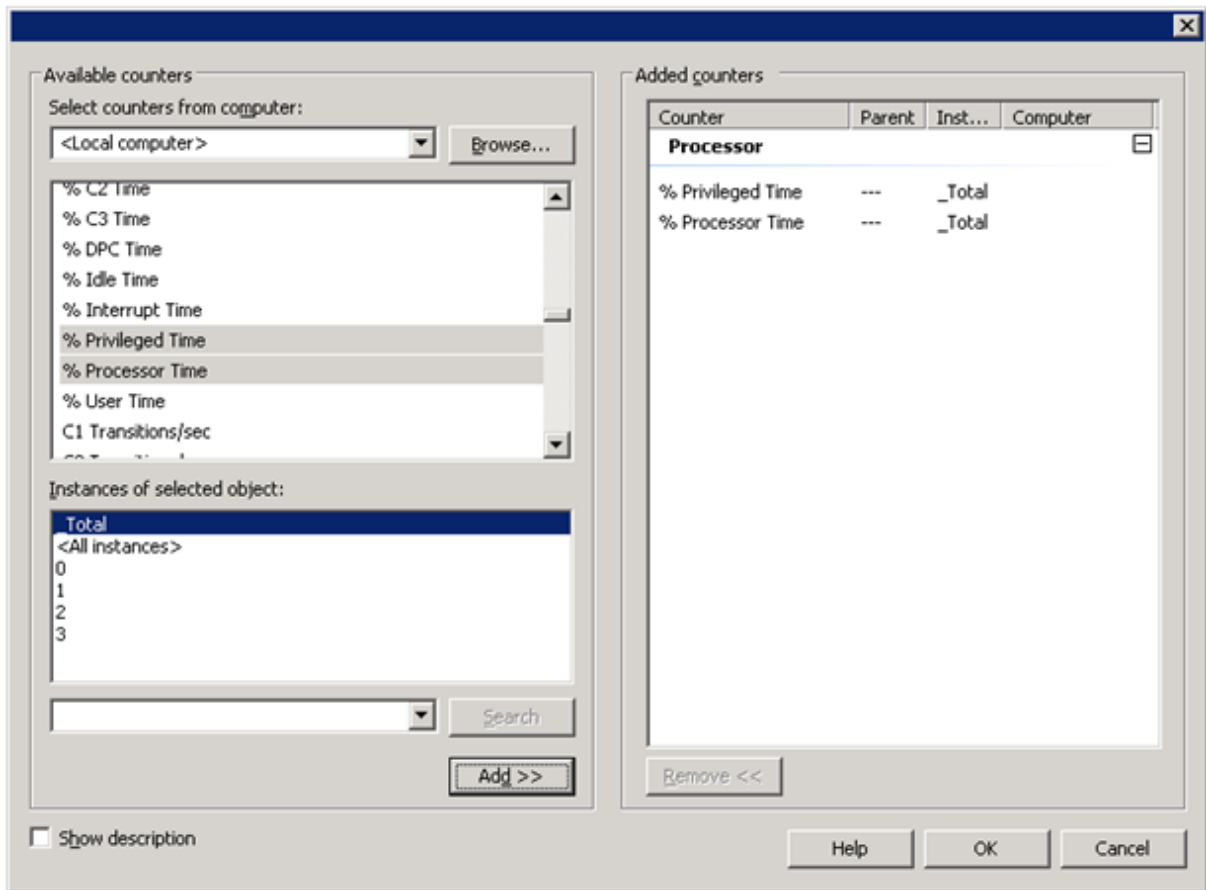
Object	Counter	Description	Comment / Values
Memory	Pages / Second	Page Faults/sec is the average number of pages faulted per second. It is measured in number of pages faulted per second because only one page is faulted in each fault operation, hence this is also equal to the number of page fault operations. This counter includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory.) Most processors can handle large numbers of soft faults without significant consequence. However, hard faults, which require disk access, can cause significant delays.	50 to 300 / second.
Disk	% Disk Time	% Disk Time is the percentage of elapsed time that the selected disk drive was busy servicing read or write requests.	Even a consistent disk usage of 5% can adversely affect performance.  Anything above 50 to 80% needs resolution.
Disk	Avg. Disk Queue Length	Avg. Disk Queue Length is the average number of both read and write requests that were queued for the selected disk during the sample interval.	Average value < 2 per disk spindle.
Disk	Avg. Disk sec/Read	Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.	3-10 milliseconds. Occasional spikes are ok but consistent 3-10 millisecond or higher can be a problem for SQL files like mdf and ldf.
Disk	Avg. Disk sec/Write	Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.	3-10 milliseconds. Occasional spikes are ok but consistent 3-10 millisecond or higher can be a problem for SQL files like mdf and ldf.
Disk	Disk Write Bytes/sec	Disk Write Bytes/sec is rate at which bytes are transferred to the disk during write operations.	Investigate high values.
Disk	Disk Read Bytes/sec	Disk Read Bytes/sec is the rate at which bytes are transferred from the disk during read operations.	Investigate high values.
Network Interface	Bytes Total/sec	Bytes Total/sec is the rate at which bytes are sent and received over each network adapter, including framing characters. Network Interface\Bytes Total/sec is a sum of Network Interface\Bytes Received/sec and Network Interface\Bytes Sent/sec.	Average value should be < 50% of NIC capacity
SQL Server : Buffer Manager	Buffer cache hit ratio	Percentage of pages that were found in the buffer pool without having to incur a read from disk.	Average value >= 90%
SQL Server : Memory Manager	Target Server Memory (KB)	Total amount of dynamic memory the server is willing to consume	It will be close to size of physical memory or if we have throttled the value from SQL Management studio.

Object	Counter	Description	Comment / Values
SQL Server : Memory Manager	Total Server Memory (KB)	Total amount of dynamic memory the server is currently consuming	It should be close to Target Server Memory.
SQL Server : Statistics	Batch Requests / Sec	Number of SQL batch requests received by server.	No particular threshold. Depending on SQL Servers hardware specs it can handle higher values.  It indicates the load on the SQL Server. Remember no two batch requests are same. So it's not correct to assume they do the same amount of work.

## Perfmon Setup

Use the following steps to set up the Perfmon and add counters to it.

1. From the **Windows** desktop, click the **Start** button and select **Run**.
2. The **Run** window displays. In the **Open** field, enter Perfmon.
3. Click **OK**.  
The **Reliability and Performance Monitor** window appears.
4. Expand the **Data Collector Sets** node.
5. Right-click on the **User Defined** node. From the context menu, select **New>Data Collector Set**.  
The **Create New Data Collector Set Wizard** displays.
6. In the **Name** field, enter Epicor Performance Diagnostics.
7. Select the **Create manually (Advanced)** radio button option.
8. Click **Next**.
9. Select the **Create data logs** radio button option.  
This activates a series of check boxes.
10. Select the **Performance** counter check box.
11. Click **Next**.
12. You are now ready to add the specific counters you wish to monitor. Click the **Add** button.  
A new window displays, listing all of the available counters.



13. Select a counter you wish to add. In this example, you select both the % Privileged Time and % Processor Time counters.
14. Click the **Add** button.  
The selected counters appear within the **Added counters** list.
15. Click **OK**.  
You return to the Create New Data Collector Set Wizard.
16. Change the **Sample interval** to reflect how often you want the counter to check the performance of this aspect of the system. For example, you could select a Sample Interval of six seconds, indicating you want the Perfmon to check the %Privileged Time and % Processor Time counters to be checked every six seconds.
17. Click **Next**.
18. Now use the **Root directory** field to define where you want the log file to be saved. You can enter this directory path or click the **Browse** button to find and select it.
19. Click the **Save and close** radio button option.
20. Click **Finish**.  
You return to the Reliability and Performance Monitor window.

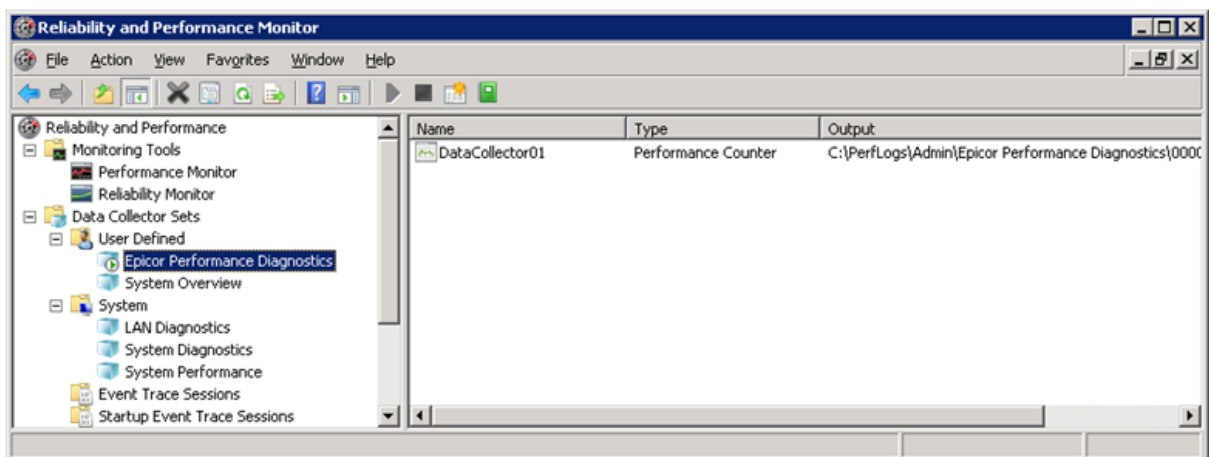
## Define Data Collectors Set Properties

Use the following steps to define the properties for your data collectors set.

Before you can define the data collectors set properties, you need to indicate which counters you will monitor through the Perfmon tool. Review the previous Perform Set section for information.

1. From the **Reliability and Performance Monitor** window, expand the **Data Collectors Set** node and the **User Defined** node.
2. Right-click on the **Epicor Performance Diagnostics** data collector set. From the context menu, select **Properties**.  
The **Epicor Performance Diagnostics Properties** window displays.
3. Click on the **Directory** tab.
4. Verify the **Root** directory value points to the location you need. You can accept the default value, or click the **Browse** button to find and select a different directory path.
5. Click the Reliability and Performance Monitor window tab.
6. Click the **Add** button to create a new schedule. For example, you can create a schedule that runs every day starting at 12:00 AM.
7. Now click on the **Stop Condition** tab. You use the parameters on this tab to indicate when you want the Perfmon tool to stop collecting data.
8. Select one of the **Limits** options. For example, you could select the Maximum Size check box and enter a value, like 1024 MB, that indicates how large the log file can become. Once the log file exceeds this size, the Perfmon tool no longer places data into the log.
9. Click **OK**.

You return to the Reliability and Performance Monitor window. This window will look similar to the following illustration.



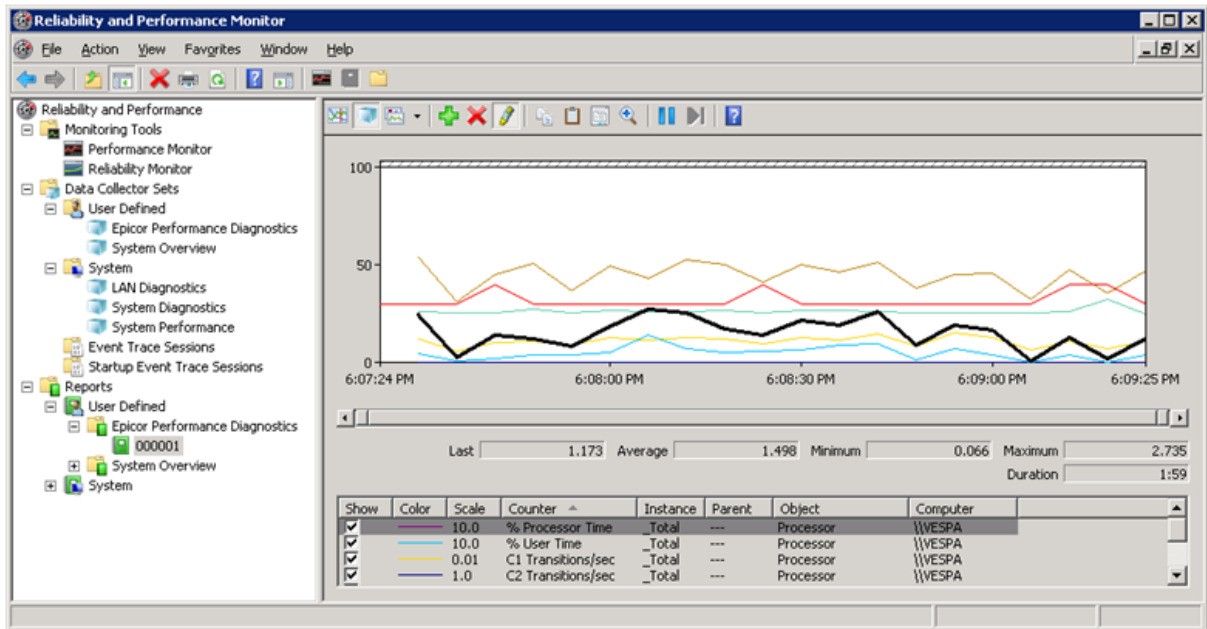
## View Report

After you collect the performance data, do the following to view the report.

Use this information to evaluate the performance analysis. You can also send the generated **.blg** file to Epicor Technical Support to review your system performance.

1. From the **Reliability and Performance Monitor**, expand the **Reports** node and the **User Defined** node.
2. Expand the **Epicor Performance Diagnostics** node.
3. Select the report you want to review.

The report displays in the Reliability and Performance Monitor. It will look similar to the following illustration.





## Index Enhancement Scripts

---

If you use SQL or SQL Unicode, use the following steps to verify that each of the following SQL performance script has been properly applied:

- **904-500DF.sql**
- **Epicor.9045XX.alter.index.allow\_page\_locks.sql**
- **Unique\_Index.9045XX.sql**

These scripts may need to be run against your database to enhance the performance.



**Important** The index enhancement scripts do not need to be run on Epicor 9.05.700 and later versions. The scripts and the procedures in this section apply only to Epicor versions prior to 9.05.700.

### Verify Script: 904-500DF

---

If the script hasn't been applied, the "SQL Scripts need to be run" message may appear when you try to log into your Epicor application.

1. To test that the script has been applied, open SQL Studio and click on your database to select it.
  - a. Expand the nodes: **<database>/ Programmability / Functions / Scalar-Valued Functions**
  - b. If the **dbo.StringTime** query exists, your SQL script was properly applied. You can skip to the next verification. If it does not exist, continue with step 2.
2. Click on your database to select it.
3. Click the **New Query** button. A query window opens.
4. Go to the location where your SQL scripts are located. On a standard installation, the SQL scripts are located in `\Epicor Software\Epicor905\ClientServer\db\SQL`.
5. Click on the **904-500DF.sql** file to select it. Drag and drop the file into the open query window. Click the Execute button. This process may take several minutes. The message "Command completed successfully" displays.
6. To test that the script has been applied, open SQL Studio and click on your database to select it.
  - a. Expand the nodes: **<database>/Programmability/Functions/Scalar-Valued Functions**
  - b. The **dbo.StringTime** query should now be listed.

## Verify Script: Epicor.9045XX.alter.index.allow\_page\_locks.sql

---

If the script hasn't been applied, the "SQL Scripts need to be run" message may appear when you try to log into your Epicor application.

1. To test that the script has been applied, open SQL Studio and click on your database to select it.
  - a. Expand the nodes: **<database>/ Tables / dbo.PDMMMap / Indexes / PDM**.
  - b. Right-click on PDM and select Properties.
  - c. Click **Options**. If the **Use page locks when accessing the index** check box is clear, your SQL script was properly applied. You can skip to the next verification. If the check box is selected, continue with step 2.
2. Click on your database to select it.
3. Click the **New Query** button. A query window opens.
4. Go to the location where your SQL scripts are located. On a standard installation, the SQL scripts are located in `\Epicor Software\Epicor905\ClientServer\db\SQL`.
5. Click on the **Epicor.9045XX.alter.index.allow\_page\_locks.sql** file to select it. Drag and drop the file into the open query window. Click the **Execute** button. This process may take several minutes. The message "Command completed successfully" displays.
6. To test that the script has been applied, open SQL Studio and click on your database to select it.
  - a. Expand the nodes: **<database>/ Tables / dbo.PDMMMap / Indexes / PDM**
  - b. Right-click on PDM and select **Properties**.
  - c. Click **Options**. The **Use page locks when accessing the index** check box should now be clear.

## Verify Script: Unique\_Index.9045XX.sql

---

If the script hasn't been applied, the "SQL Scripts need to be run" message may when you try to log into your Epicor application.

1. To test that your SQL script has been applied, open SQL Studio and click on your database to select it.
  - a. Expand the following nodes: **<database>/ Tables / dbo.abccode / Indexes / abccode#\_#progress\_recid**
  - b. Right-click on `abccode#_#progress_recid` and select **Properties**.
  - c. Click **Options**. If the **Ignore duplicate values** check box is clear, your SQL script was applied properly. You can skip to the next verification. If the check box is selected, continue with step 2.
2. Click on your database to select it.

3. Click the **New Query** button. A query window opens.
4. Go to the location where your SQL scripts are located. On a standard installation, the SQL scripts are located in `\Epicor Software\Epicor905\ClientServer\db\SQL`.
5. Click on the **Unique\_Index.9045XX.sql** file to select it. Drag and drop the file into the open query window. Click the **Execute** button. This process may take several minutes. The message "Command completed successfully" displays.
6. To test that the script has been applied, open SQL Studio and click on your database to select it.
  - a. Expand the following nodes: **<database>/ Tables / dbo.abccode / Indexes / abccode#\_#progress\_recid**
  - b. Right-click on **abccode#\_#progress\_recid** and select **Properties**.
  - c. Click **Options**. The **Ignore duplicate values** check box should now be clear.
7. Close SQL Studio.

## Additional Performance Troubleshooting

---

This section documents actions you can take to reduce poor performance within both your overall system and the application.

### System Troubleshooting

---

- Epicor recommends that the **\Epicor** folder be shared Read Only for everyone that accesses the server.
- If the **Windows XP** or **Windows 7** firewall is active, it may slow performance. Test this firewall against other firewall options to discover which one reduces performance the least but still effectively protects your system.
- Frequent virus scans on the **Mfgsysdata** and the client folders slow performance. Schedule virus scans during off hours.
- Backing up the database during the day slows performance. Schedule database backups during off hours.
- Avoid directly accessing tables through **ODBC** or **SQL Explorer**. If you do this, records can lock-up.
- Avoid setting the **DSN** value to the wrong default isolation level within ODBC. If you do this, records can lock up.
- When multiple applications run at the same time, performance slows. Try to only use the programs you need when you need them. Close them if they are no longer active. If external companies are not being used, delete them.
- Avoid creating several **Business Process Management** (BPM) procedures that generate email messages at the same time (synchronously). If these procedures run frequently, performance is reduced. Instead, set up your BPM procedures to run asynchronously.
- Avoid activating too many change logs within the **Business Activity Manager**. Only run change logs for the specific fields you need. When too many logs are running, they slow performance. Performance slows when several large reports generate at the same time. To resolve this, move your report generation processes to separate AppServers.

### Application Troubleshooting

---

- Be sure to disable all schedules before you move from the training database to your live database. If you do not, two MRP processes run at the same time, slowing down performance.
- Epicor recommends that you do not run MRP processing during the work day. Schedule MRP processing during off hours.
- Be sure that your Business Activity Queries (BAQs) are properly written. If a BAQ pulls unnecessary data or a calculation is improperly constructed, each time this BAQ is run it slows performance.
- Avoid creating several Business Process Management (BPM) procedures that generate email messages at the same time (synchronously). If these procedures run frequently, performance is reduced. Instead, set up your BPM procedures to run asynchronously.
- Avoid activating too many change logs within the Business Activity Manager. Only run change logs for the specific fields you need. When too many logs are running, they slow performance.
- Performance slows when several large reports generate at the same time. To resolve this, move your report generation processes to separate AppServers.

- Likewise, when multiple applications run at the same time, performance slows. Try to only use the programs you need when you need them. Close them if they are no longer active.

# Index

## A

- add files 16
- analyze client logs 10
- analyze server logs 15
- application server dotR installation 8
- application troubleshooting 52

## B

- baseline network/server performance 23
- baseline performance test 22
- bpm directives 39

## C

- clear results 11, 16
- clear selected 16
- client cache 25
- client customizations 39
- client diagnostic scenarios 12
- client diagnostics 9
- client logs, analyze 10
- client logs, generate 9
- client trace path 11
- common patterns 40
- configuration settings file 27
- custom format 11
- customer retrieval 28

## D

- data collectors set properties 47
- developer mode 39
- dotR installation, application server 8

## E

- e9 performance diagnostic tool 39
- e9 performance diagnostic tool 22, 23
- epicor performance diagnostic tool 7
- epicor technical support 41
- epicor version 17
- everything is slow 41
- exclude system calls 11
- export to excel 12, 17

## F

- fields client diagnostics 11
- fields server diagnostics 16
- frequent user complaints 41

## G

- generate client logs 9
- generate diagnostics 12, 17, 28, 29, 31
- generate server logs 13
- getrowskeepidle 22

## I

- ignore getrowskeepidletime 17
- installation epicor performance tool 7

## L

- lan 22
- log errors sheet 17
- log file date format 12

## M

- millisecond threshold 17

## N

- network diagnostics 18
- network parameters 18
- network test 20

## O

- open sales order entry form 25

## P

- perfmon 42
- perfmon counter list 43
- perfmon setup 45
- performance diagnostics 39
- performance troubleshooting 52
- purchase order entry performance 31
- purchase order entry test 31

## R

- reboot corrects poor performance 42
- reliability and performance monitor 48
- results sheet 12, 17
- running processes sheet 17

## S

- sales order detail line performance 29
- sales order line entry test 29
- server customizations 39

- server diagnostic scenarios 18
- server diagnostics 13
- server logs, analyze 15
- server logs, generate 13
- sheets client diagnostics 11
- sometimes everything is slow 41
- specific issues 39
- specific program is slow 41
- start the test 22
- summary sheet 12, 17
- system troubleshooting 52

## **T**

- test form performance time 26

- test procedure 23
- test setup 22
- trace file 40
- trace log 23, 28, 29, 31, 41
- troubleshooting list 33

## **U**

- unable to reproduce 42

## **V**

- view report 48

# EPICOR.

Additional information is available at the Education and Documentation areas of the EPICweb Customer Portal. To access this site, you need a Site ID and an EPICweb account. To create an account, go to <http://support.epicor.com>.