FINAL DRAFT

MSGovern

Govern Validator

Release 5.1 Version 1

Last Revision Update: 9/27/2013

FINAL DRAFT

MS Govern

Govern Validator

Version: 1.0

September 2013 - Release 5.1

MSGovern

Disclaimer

MS Govern has taken due care in preparing this manual. However, nothing contained herein modifies or alters in any way the standard terms and conditions of the purchase, lease, or license agreement by which the product was acquired, nor increases in any way the liability of MS Govern to the customer.

Preface

Welcome to Govern for Windows, a comprehensive and fully integrated transaction-driven system written exclusively for local governments. Govern includes a wide variety of database modules:

Computer-Assisted Mass Appraisal (CAMA)

- Appeals & Grievances
- Appraisals / Property Valuations
- Comparables Sales Management

Financial Management

- Account Receivable
- Cash Collection

Land Management

- Business & Individual Licenses
- Complaint Tracking
- Leasing
- Permit Tracking & Inspection Scheduling
- Planing
- Violations

Revenue Management

- Aircraft & Boat Excise Tax
- Miscellaneous Billing
- Personal Property Tax Billing
- Real Property Tax Billing
- Self-Reported Tax Billing
- Special Assessments
- Tax Title / Tax Lien / Tax Sales
- Utility Billing

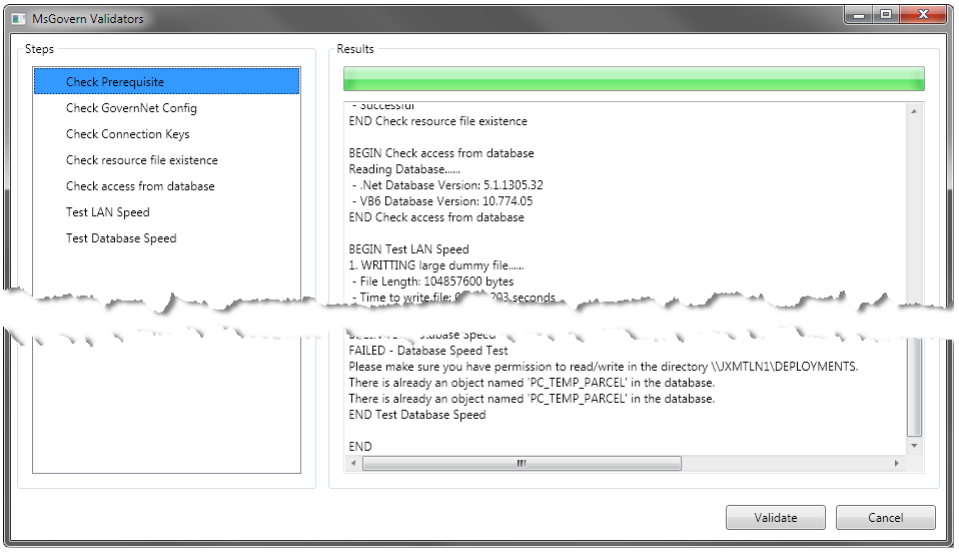
Since 1980, MS Govern, has worked hand-in-hand with State and Local Governments to simplify the implementation of software solutions that automate the flow of information related to their properties.

MSGovern

Table of Contents

The Govern Validator

Overview



In any software application or suite of applications, for proper functioning, there must be an optimum installation environment. The requirements for Governs' suite of applications are very specific, as a result the end-user can be challenged in determining whether they have the required installed applications to allow for the stable functioning of Govern.

The *Govern Validator* is a utility that will review a target system, to validate the presence or status of required components or settings. The three (3) areas of a systems environment that are reviewed are:

- The GovernNet Configuration file (*GovernNetConfig.xml settings*)
- The existence, location, and access of *Resource Files*
- Database Access
- **NEW!** A *LAN Performance Test*
- **NEW!** A *Database Performance Test* (NEW)

From the GovernNetConfig.xml file, the *Govern Validator* will determine the location of Connection Keys (CK), Resource File name, type of authentication, etc. In addition the location of the Resource file will be determined, and access

to the database. Any or all of these factors can affect the functioning of the Govern.NET application.

When to run the Govern Validator

The Govern Validator can be run under the following conditions:

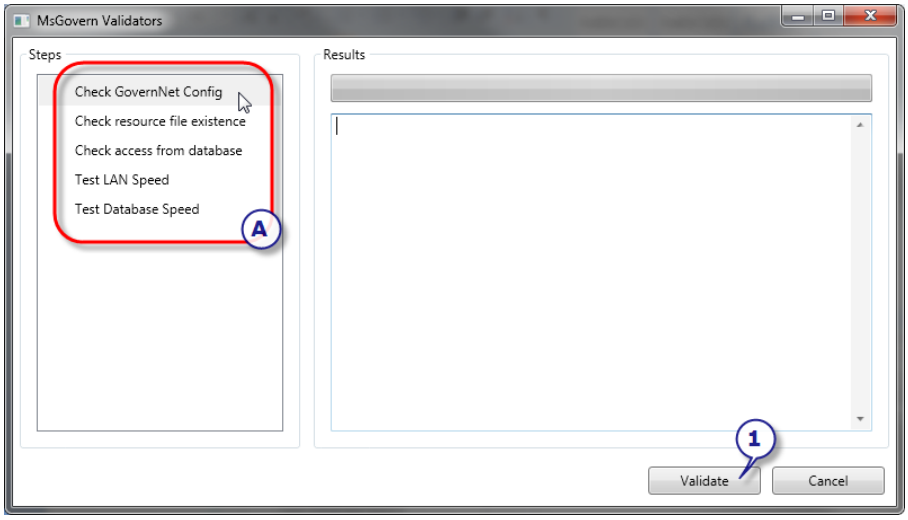
- After installation of any required Govern.NET applications, the Govern Validator may be run prior to launching any of the installed applications.
- During incidents where any of the Govern.NET applications fail to launch; the Govern Validator can be invaluable for assessing missing or wrongly configured components.
- As a verification that there are no existing issues with LAN performance.
- A means of verifying that database performance is optimal for the Govern application.

Launching the Validator

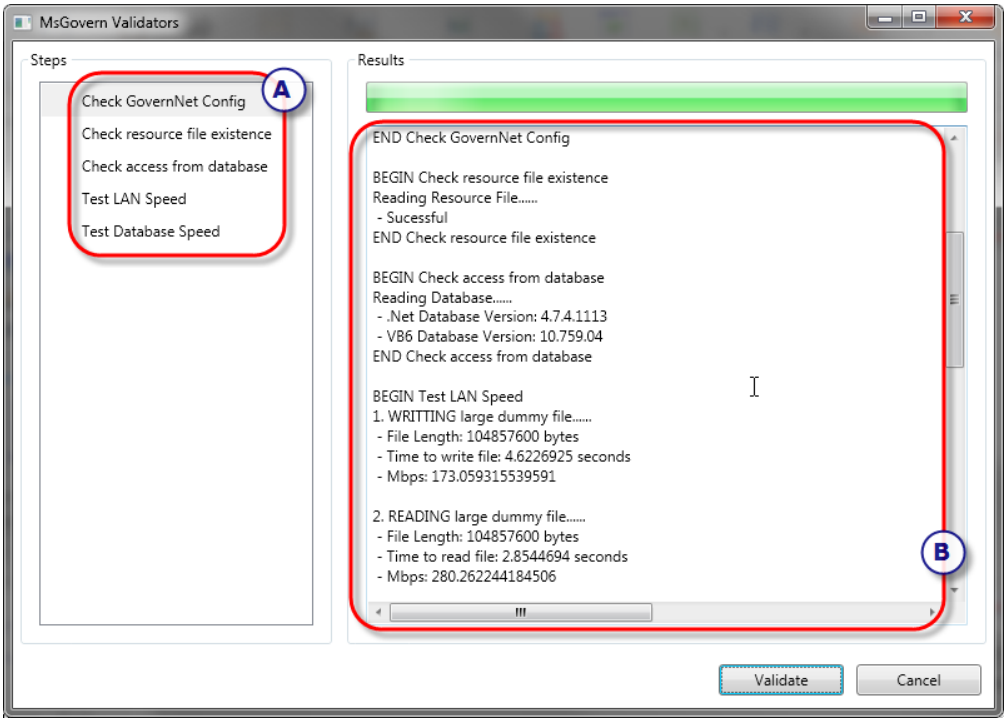
The Govern Validator is installed and initiated like any other application in the Govern.NET suite. It is installed from a Deploy EZ deployment, and can be found in the MS Govern program group.

To run the Govern Validator...

1. Launch the Govern Validator application; the application window will appear.



2. In the interface, the components that are to be reviewed are displayed in the upper left hand corner, under **Steps** (**A**).
3. Click **Validate** in the lower right hand corner of the interface (**1**).
4. The application will begin its determination, and after a length of time the results of its findings will be displayed under the **Results** section of the interface (**B**).



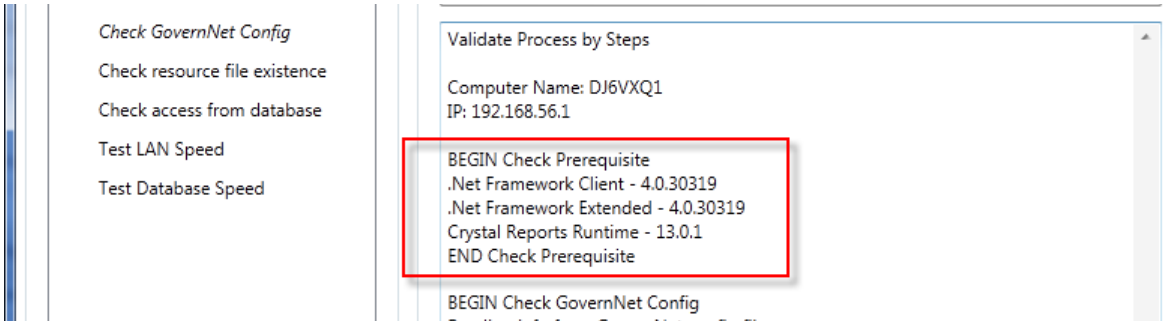
Note: The length of the determination process is dependent upon individual system configuration.

5. After analyzing the results, you can click inside the results pane to select the displayed text; click **Ctrl** + **A**, to select all text.
6. Press the **Ctrl** + **C** combination to copy the text; it can then be pasted into a text editor for further review.
7. Click **Cancel** to close the application.

Reviewing the Results of the Investigation

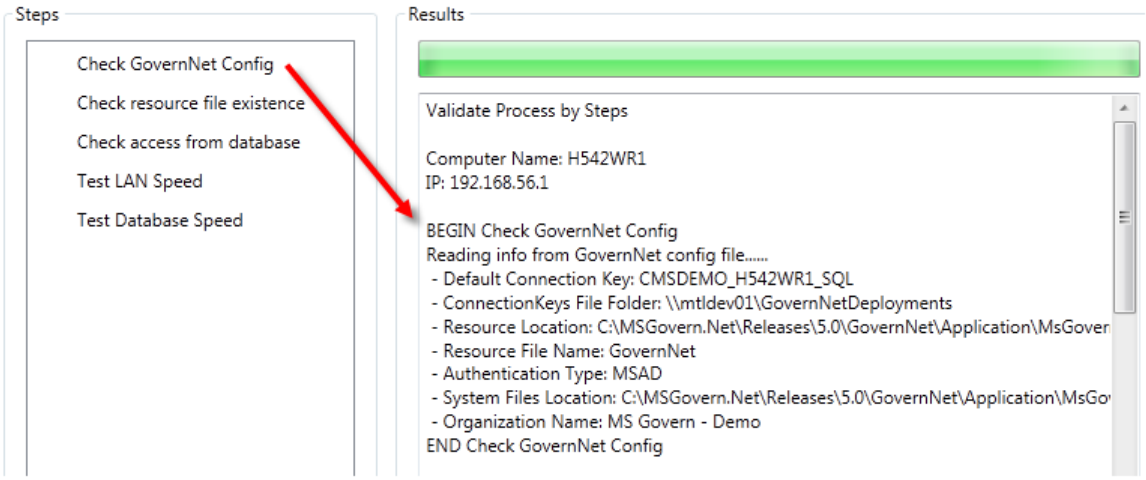
The following is a breakdown of the components that are reviewed by the validation process.

Check Prerequisite



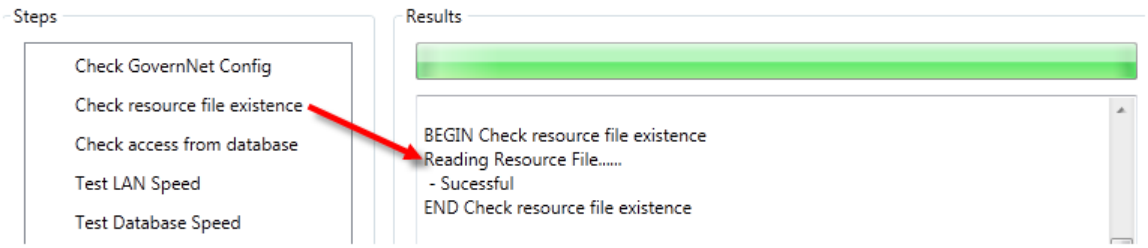
- Determine *Computer Name* and *IP Address*
- Verify versions of *.NET Framework Client*, *Framework Extended*, and *Crystal Reports Runtime*

GovernNET Config Information



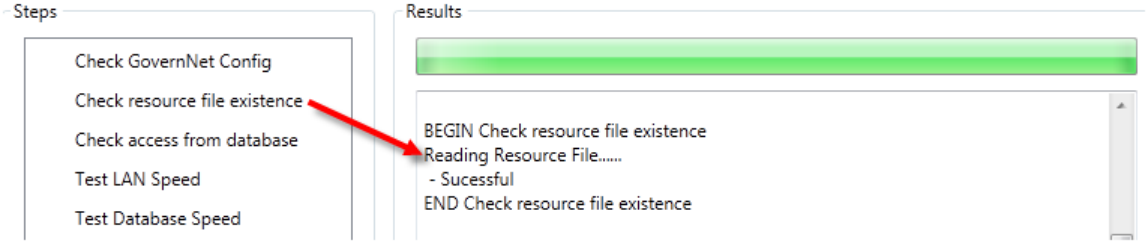
- Determines the name of the default Connection Key
- Path to *ConnectionKeys* File Folder
- Path to *Resource* file
- *Resource File* Name
- *Method of Authentication* configured in *DeployEZ*
- Location of *System File* Folder
- The *Organization Name* that the application is registered under

Check resource file existence



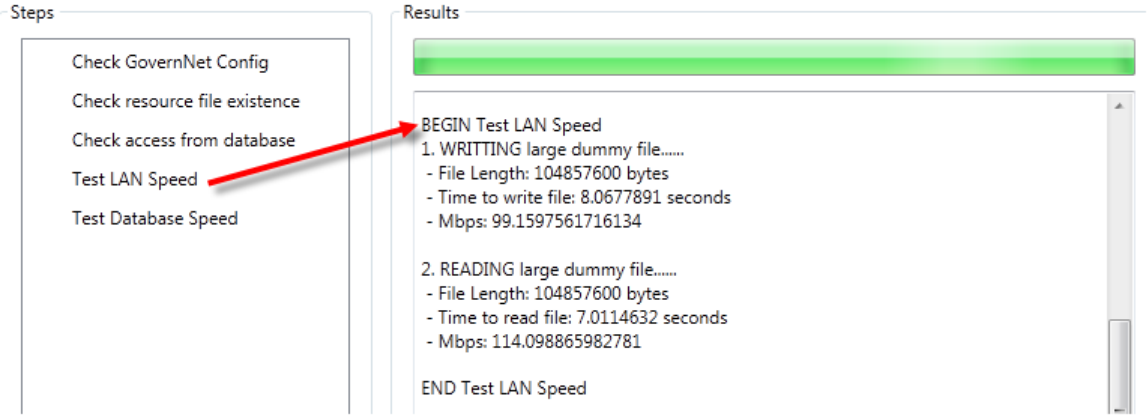
- Verification of existence of *Resource File*

Check Access to Database



- Use the Connection Key to access the Database

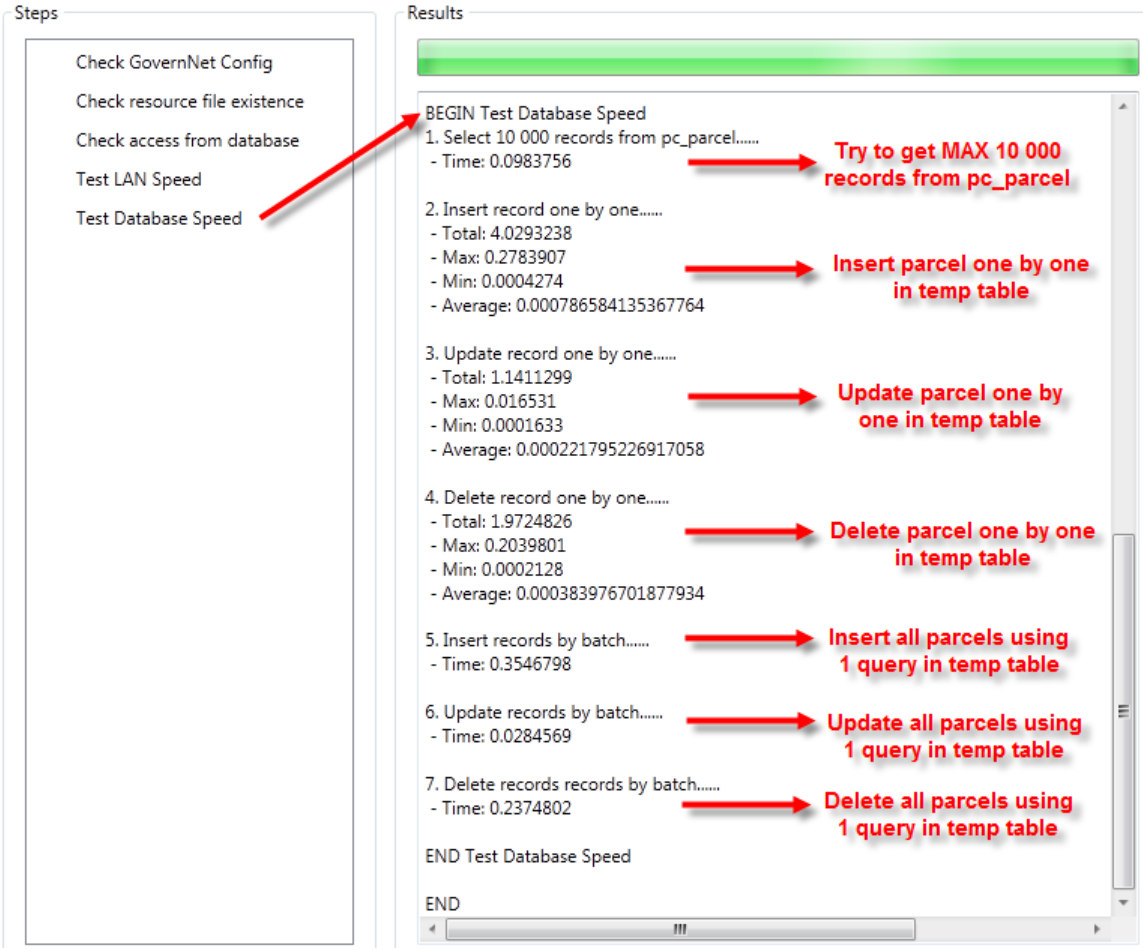
Test LAN Speed



- **NEW!** This process tests the LAN connection speed by reading and writing on the server side. A log file called **MonitorLAN.log** is created in the *Deployment Folder* on the server under the **Trace** sub-folder.

  Refer to Details of the Test LAN Speed process on page 7*. Also refer to Converting .log File to .CSV in MS Excel* on page 9 *for saving in MS Excel format.*

Test Database Speed



- **NEW!** The database speed is tested with a series of write, read, and delete operations. A log file called **MonitorDB.log** is created in the *Deployment Folder* on the server under the **Trace** sub-folder.

  Refer to Details of the Test Database Speed process on page 7 *for details. Also refer to Converting .log File to .CSV in MS Excel* on page 9 *for saving in MS Excel format.*

Details of the Test LAN Speed process

During the **Test LAN Speed** process, a central log file is created for all Deployments.

File Location

[Deployment Directory] \ Trace \ **MonitorLAN.log**

The first time that the process is run, the log file **MonitorLAN.log**, will be created. The next time the process is run, and for all other subsequent runnings of the process, the log information will be appended to the file.

Each test will create a single line in the log file; the following information is recorded:

- [**RunAt**]: Date/Time of the performance Test
- [**MachineName**]: Name of the computer executing the test
- [**IP**]: IP of the computer executing the test
- [**Write**]: Total Time for copying 100 Mbits to the deployment server
- [**Mbps**]: megabit per second for copying 100 Mbits to the deployment server
- [**Read**]: Total Time to read 100 Mbits to the deployment server
- [**Mbps**]: megabits per second for reading 100 Mbits to the deployment server.

Note: Administrators should note that the log file contains *Machine Names* and *IP* addresses that may be considered sensitive.

Details of the Test Database Speed process

During the **Database Performance Test** process, a central log file is created for all Deployments.

File Location

[Deployment Directory] \ Trace \ **MonitorDB.log**

The first time that the process is run, the log file **MonitorDB.log**, will be created. The next time the process is run, and for all subsequent runnings of the process, the log information will be appended to the file.

Each test will create a single line in the log file; the following information is recorded:

- [**RunAt**]: Date/Time of the performance Test
- [**Deployment**]: Name of the deployment used for the test
- [**MachineName**]: Name of the computer executing the test
- [**IP**]: IP address of the computer executing the test
- [**Select**]: Total time to execute a SELECT of 10 000 records on PC_PARCEL (No condition)
- [**InsertTotal**]: Total time to INSERT 10 000 records in a TEMP table (one by one)
- [**InsertMax**]: Maximum Insert time
- [**InsertMin**]: Minimum Insert time
- [**InsertAverage**]: Average Insert time
- [**UpdateTotal**]: Total time to UPDATE 10 000 records in a TEMP Table (one by one, simply update **Last_Modif_Date**)
- [**UpdateMax**]: Maximum UPDATE time
- [**UpdateMin**]: Minimum UPDATE time
- [**UpdateAverage**]: Average UPDATE time
- [**DeleteTotal**]: Total time to DELETE 10 000 records in a TEMPTable one by one)
- [**DeleteMax**]: Maximum DELETE time
- [**DeleteMin**]: Minimum DELETE time
- [**DeleteAverage**]: Average DELETE time
- [**InsertAll**]: Total time to Insert 10 000 records in a TEMP table (in one single query)
- [**UpdateAll**]: Total time to UPDATE 10 000 records in a TEMP table (in one single query)
- [**DeleteAll**]: Total time to DELETE 10 000 records in a TEMP table (in one single query)

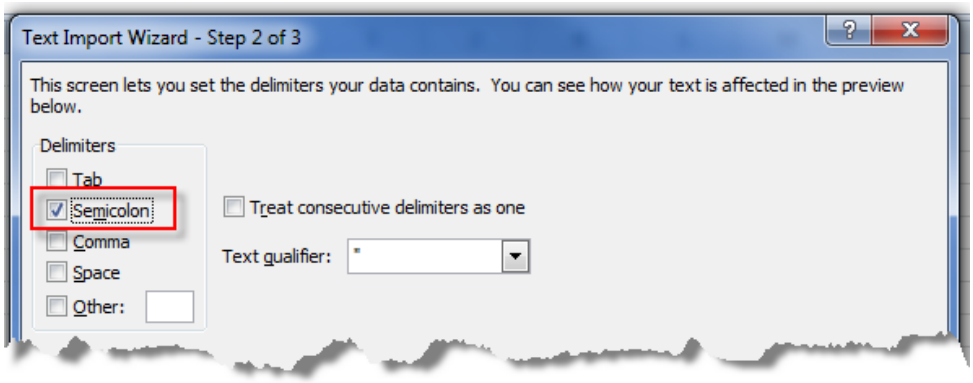
Note: Administrators should note that the log file contains *Machine Names* and *IP* addresses that may be considered sensitive.

Converting .log File to .CSV in MS Excel

The log file is a **.CSV** format file; the data is delimited with semicolons. The log file can be imported into *Microsoft Excel* and saved off as a spreadsheet for further review.

To import the log file into *Microsoft Excel*, follow the steps used for importing a .CSV file.

1. In an empty spreadsheet in MS Excel, select the **Import Text File** option.
2. At the prompt select the option for a **Semi-Colon** delimited file.



Note: When importing the file, note that the file extension is .log, therefore select the **All Files (*.*)** option to view the **MonitorLAN.log** file.

Once imported, the file can be formatted and saved as an MS Excel **.xls/.xlsx** format file.

Conclusion

The results displayed by the *Govern Validator* can be instrumental in trouble-shooting applications that will not launch, or display incorrect information. In addition the LAN and Database speed tests can provide insight into environments where there may be application performance issues.