Eurosoft
computer reliability solutions

Pc-Check® Windows® User Guide

Windows®-based Computer Diagnostic Software

Assuring Computer Service Reliability

www.eurosoft-uk.com

Technical Support

Eurosoft (UK) Ltd.

Head Office
3 St. Stephen's Road
Bournemouth
Dorset BH2 6JL
United Kingdom
Tel +44 (0)1202 297315
Fax +44 (0)1202 558280
Email: support@eurosoft-uk.com
Website: www.eurosoft-uk.com

Eurosoft (US) Inc.

Sales Office
706 Jackson Street
Sioux City, IA 51105
United States of America
Tel +1 (888) 980 9595
Fax +1 (866) 615 9384
Email: info@eurosoft-us.com
Website: www.eurosoft-us.com

Support

Europe	Email: support@eurosoft-uk.com
United States	Email: support@eurosoft-us.com

Overview

This document has been written to give information about features and configuration available with Eurosoft Pc-Check Windows.

All XML input and output files must conform to the standard as defined at http://www.w3.org/TR/REC-xml/

Examples of XML output may not exactly match your output for this version. They are only supplied as syntactical examples.

Table of Content

Understanding Pc-Check Windows Testing Architecture

Before continuing, it is important to understand how Pc-Check Windows Testing Architecture is constructed.

In order to provide as much flexibility as possible to its users, Pc-Check Windows is architected into four parts, each of them can be changed separately to suit user needs:

1. Base Pc-Check Windows – this consists of main user interface executable, diagnostic test modules, extra modules required for test modules and documentation files.

2. Configuration files – it is possible to change the way Pc-Check Windows works to suit user needs, for example setting a special naming pattern for result files that reflect your organizational needs or setting the output directory where these result files are stored. Pc-Check Windows application contains a menu item to create these configuration files as described in next section of this document. A default configuration file called Eurosoft_Config.xml is supplied under PcCheckWindows folder.

3. Test scripts – Although it is possible to execute diagnostic tests manually one after another, most users and organizations will prefer to have predefined test sequences that are executed without user intervention. For example, an organization may devise a specific sequence of tests to be executed if a Display Adapter problem is suspected and require all technicians to execute this sequence as part of their troubleshooting process. Pc-Check Windows application offers means to create and save test sequences as Test Scripts to be used later. Default test scripts supplied are located under:
   \Program Files (x86)\Eurosoft (US) Ltd\WinPEImageCreator\EurosoftApps\PcCheckWindows\Scripts\Diagnostics.

4. Command files – Pc-Check Windows uses command files to tie together above mentioned base application, configuration files and test scripts to automate specific testing sequences. These command files are regular executable Windows cmd files that can be edited and changed by the users using a text editor. Pc-Check Windows offers means to edit and change some parameters of command files. A default command file called PcCheckWindows.cmd is supplied under PcCheckWindows folder.

The three parts making up a specific test sequence are tied together in the command file in a command that looks similar to below example:

PcCheckWindows.exe /DF MyCompany_Config.xml /R Scripts\Diagnostics\DisplayAdapterTest.xml

This sample line starts the base application (PcCheckWindows.exe) with settings specified in MyCompany_Config.xml configuration file and immediately starts test sequence set in DisplayAdapterTest.xml file located under Scripts\Diagnostics folder. This sample line could be saved as part of a command file named MyDisplayAdapterTest.cmd under Pc-Check Windows folder.

As shown in the above sample, organizations and users can create multiple configuration files and test scripts and tie them together in multiple command files to adequately describe their test sequences.
It is advised to use default folders to save test scripts, configuration files and command files. In case you may want to save into different folders, please make sure you are pointing to correct paths in your files.

Getting Started with Pc-Check Windows

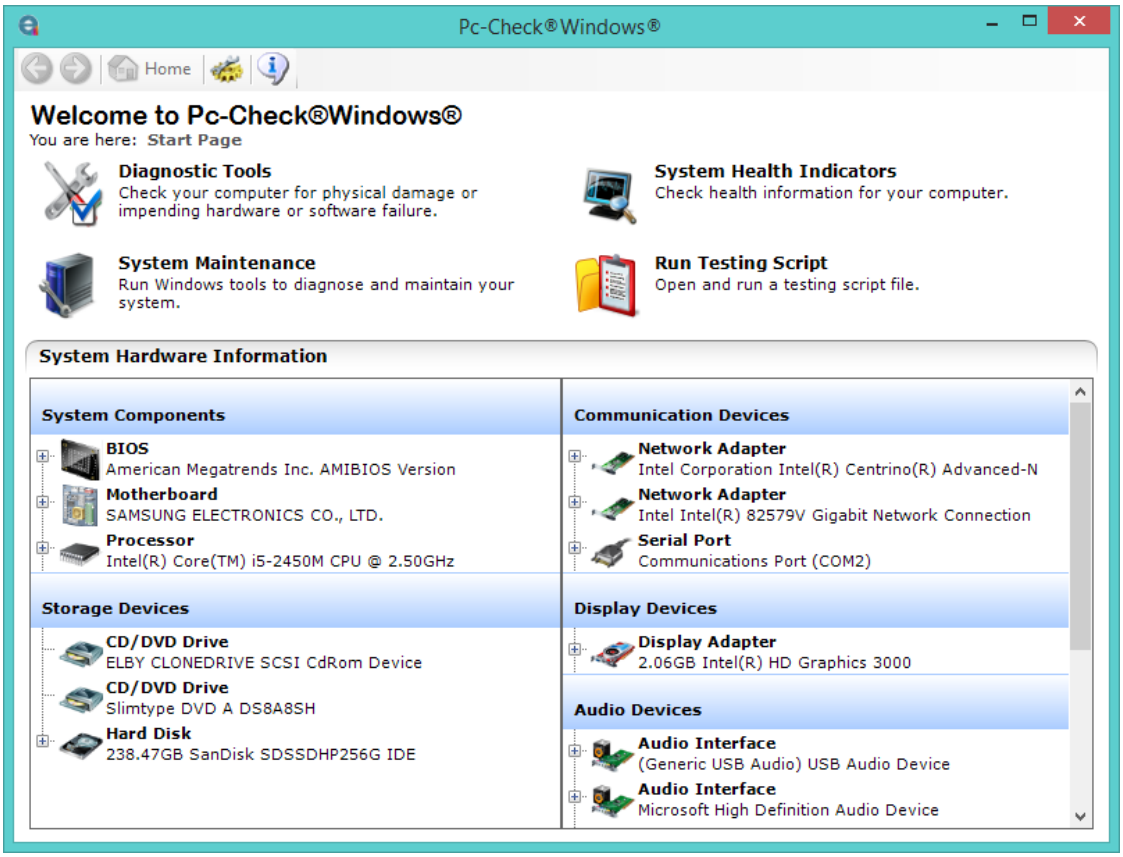
The default Startnet.cmd files used while creating deployment images ensure that Pc-Check Windows is started after Windows PE starts. If you make any changes to Startnet.cmd file to disable this behavior, please act accordingly to browse to Pc-Check Windows location.

If you want to use Pc-Check Windows under full Windows operating system, you have to insert the bootable media (USB Drives or CD/DVD media created by burning ISO image to disks), browse to the media using File Explorer, then browse to MyApplications\PcCheckWindows folder and double click to execute required command file that fits the task at hand or double click on PcCheckWindows.exe file to start the application interactively.

When you start Pc-Check Windows it will start up in the Home screen. From this screen you can easily navigate to any diagnostic or system maintenance task.

The following is a description of the elements contained in Pc-Check Windows.

Pc-Check Windows Home Screen



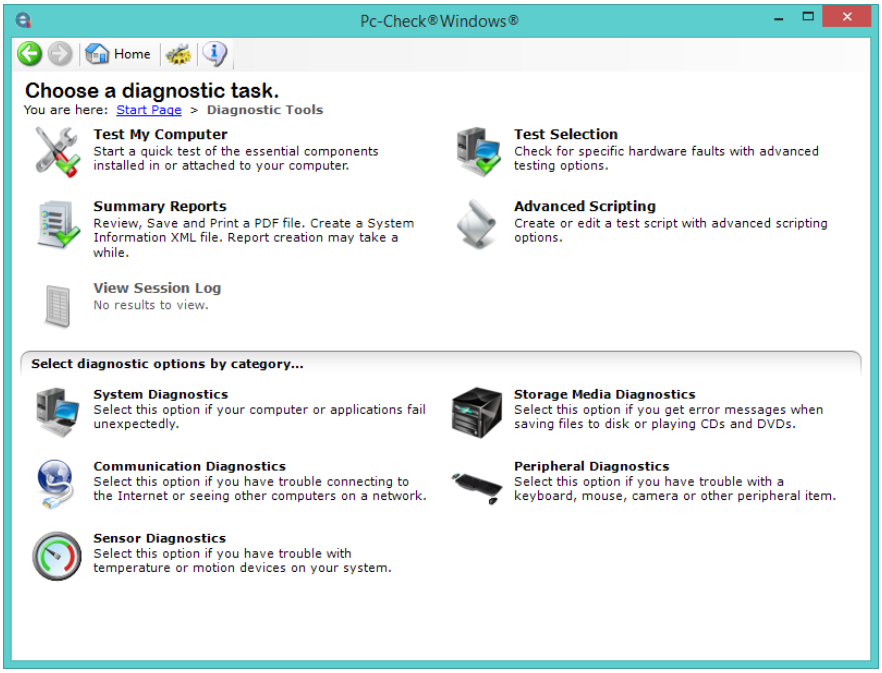
The Home screen can be accessed at any time with the Home button on the menu bar.

On the Home screen the **System Hardware Information** tab shows a listing of all of the detected components and component attributes.

Component details are accessed by clicking the + (plus symbol) to expand the item or by double clicking on the item.

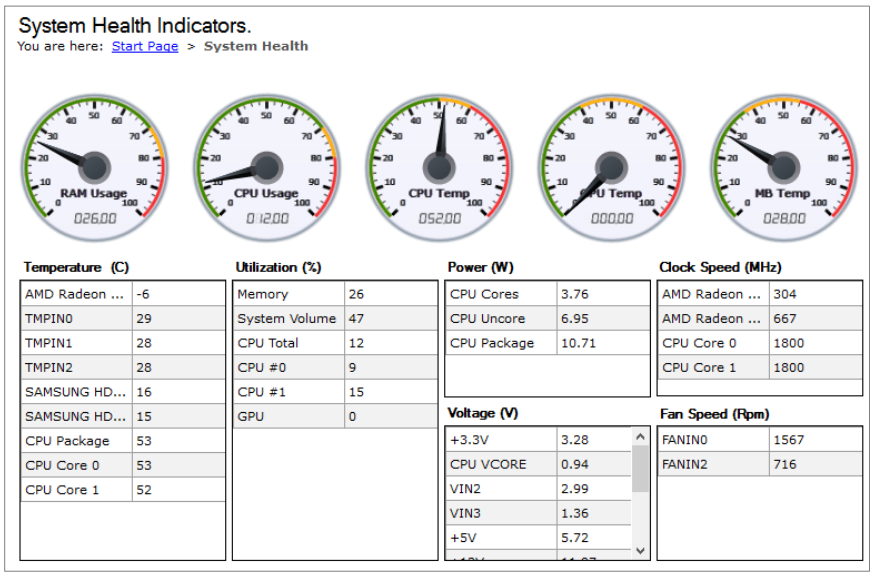
Diagnostic Tools

This menu provides diagnostic tools for testing your PC. Diagnostic testing can be quickly and selectively launched. Diagnostic Tools also provides options for saving and loading custom test scripts.



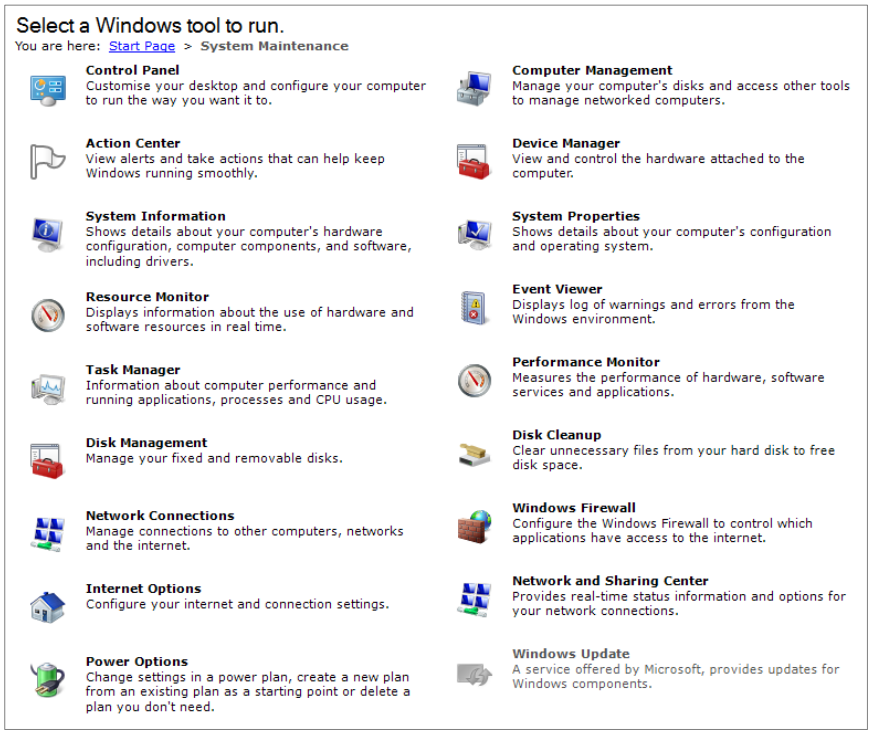
System Health Indicators

System Health Indicators provides a display panel showing real time performance indicators.



System Maintenance

This menu provides shortcuts to Microsoft Windows™ system maintenance tools including Control Panels and Computer Management.

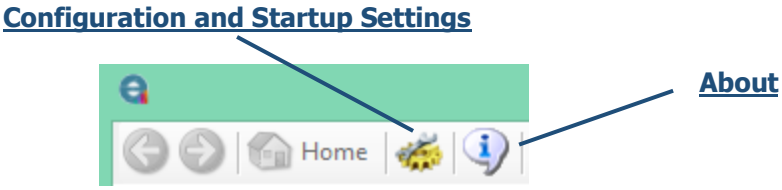


Run Testing Script

The Run Testing Script option opens a file navigation window for navigating to the script you want to run.

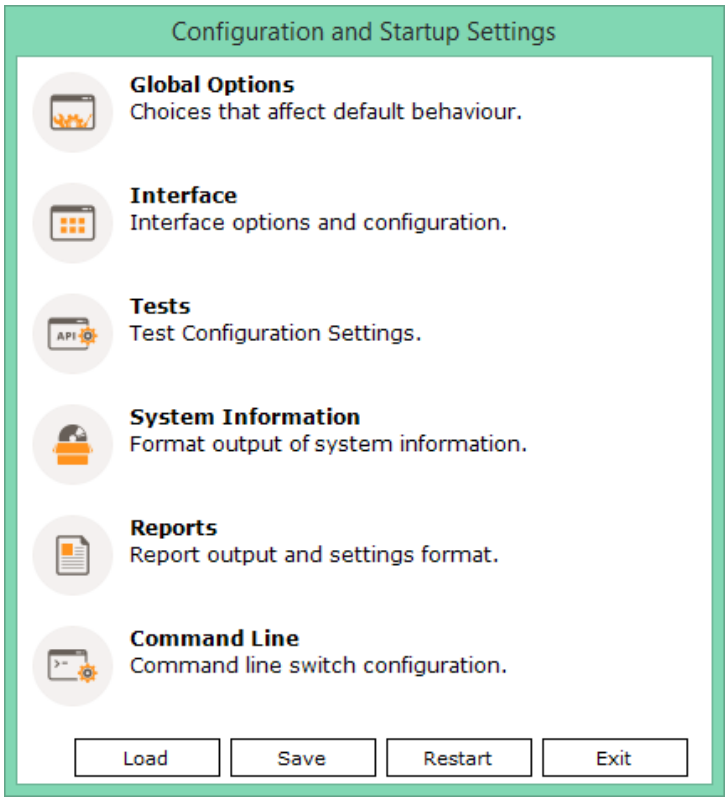
Configuration and startup settings

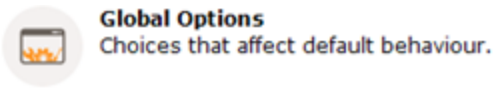
On the Home screen there are several buttons for configuring the default behavior of the program.

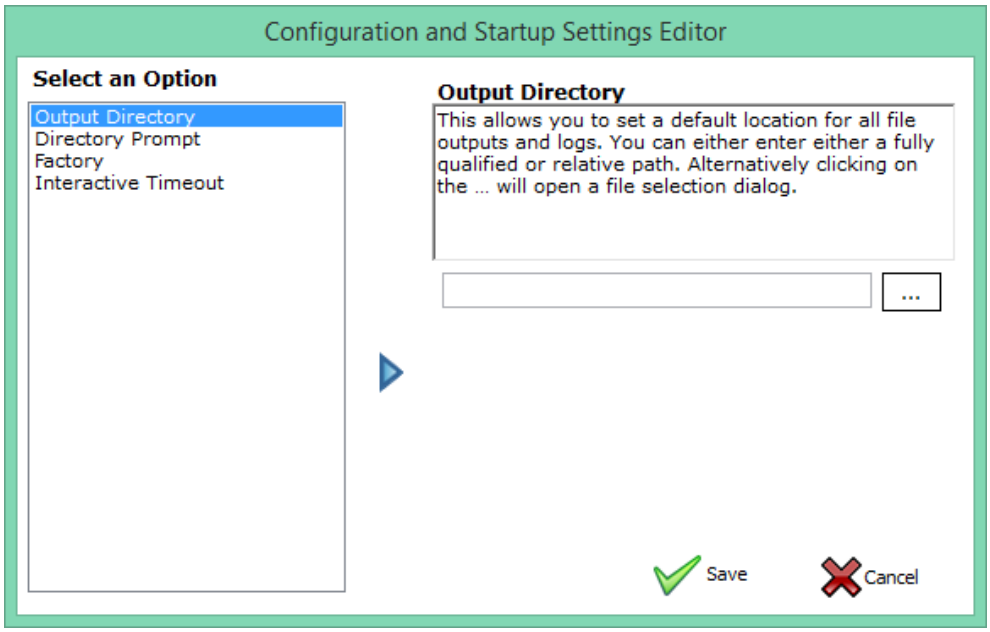


Configuration and Startup Settings

The Configuration and Startup Setting window provides options to manage all aspects of the product, including defaults, Interface, Reports, and command line options.
Each one of these options will be explained in detail.







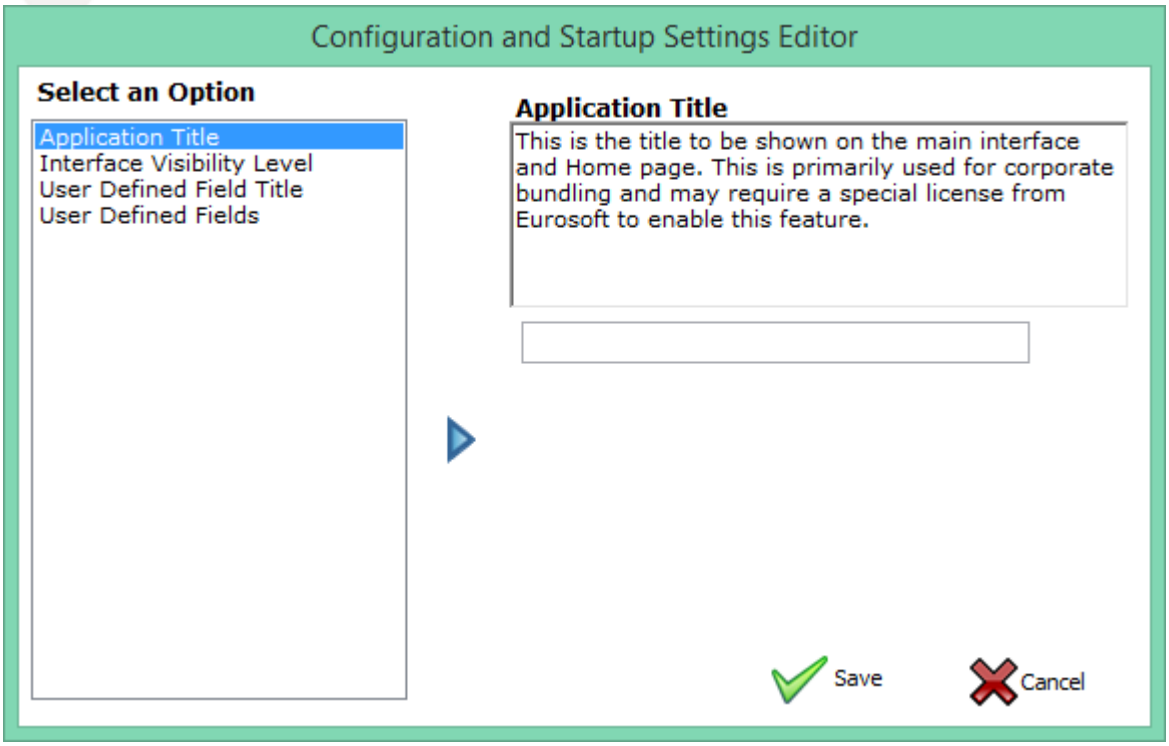
Output Directory - The output directory for Result and Sysinfo output files. This may be a relative or full file path. Click the button at the right to navigate to the file.

Directory Prompt – When this is enabled then the user will be prompted on startup to select the path for the output files to be written to.

Factory - Determines if the tests should run in interactive mode. Use checkbox to enable. In factory mode any interactive tests included in the script will not be executed.

Interactive Timeout - The length of time, in minutes, that tests or the interface should wait before reverting to non-interactive operation. Enter number of minutes in box provided.

Interface
Interface options and configuration.



Application Title – The Application title shown on the screen can be customized for example when the product is used in a corporate bundled environment. This option requires a special license from Eurosoft

Interface Visibility Level - This option controls the workflow flexibility given to the user. Use dropdown box to choose setting.

Unrestricted – No functionality is restricted and all advanced features are available. This level is most appropriate for operators who are configuring test scripts for basic technicians and advanced technicians who have no restriction on their workflow.

Partially Limited - Intermediate features are available such as basic log information. Advanced features such as script creation may be unavailable. This level is most appropriate for operators who are allowed to interpret a result from an individual test combined with any system information available.

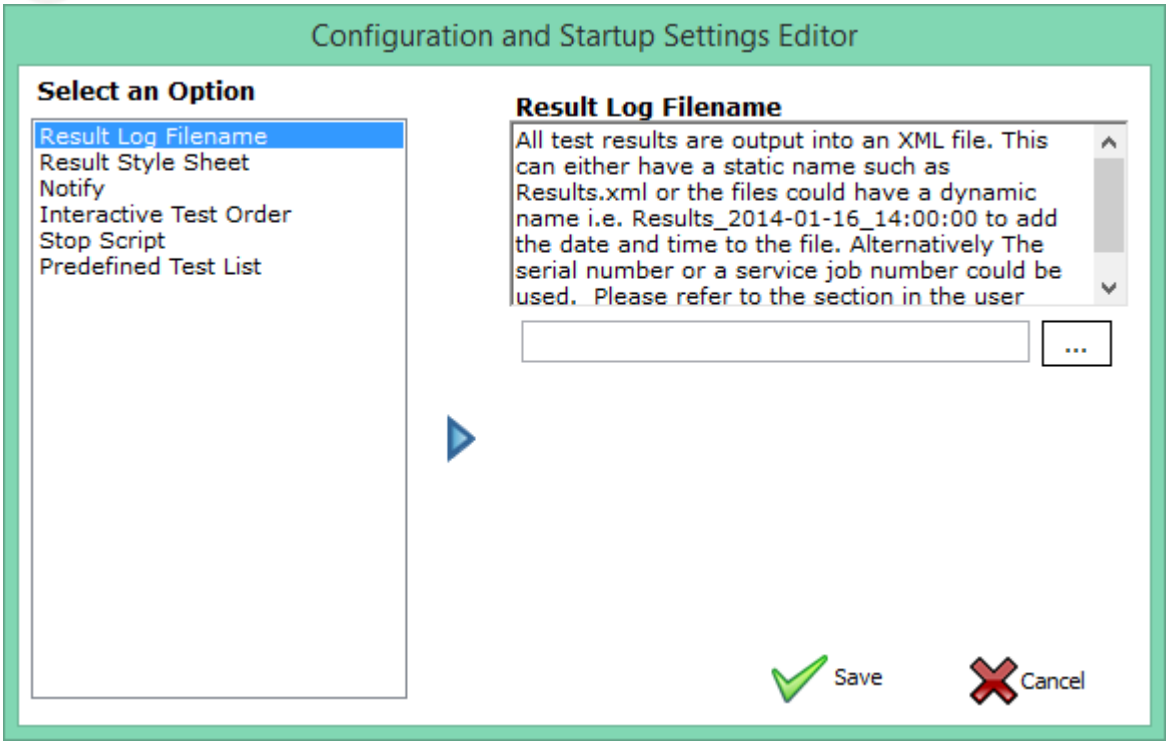
Limited - Only basic features are available. This level is most appropriate for those operators who only view the results on the interface and perform a fixed workflow depending on that result.

NOTE: Actual functionality available is application specific, the user should consult the application as appropriate.

User Defined Field Title – When user defined or prompted input is required then this is the title shown on the input screen.

User Defined Fields - List of user defined fields that will be placed in some report outputs. Complete details for User Defined Fields are listed in Appendix A. These can be used to prompt for user input, take values from environment variables, and configure output names.

Tests
Test Configuration Settings.



Result Log Filename - The filename of the output xml result log. This must have the .XML extension. Click the button at the right to navigate to the file.

This option supports user defined fields for dynamic naming, User defined fields are discussed in Appendix A.

Result Style Sheet - The style sheet to specify in the top of the result log. No checking is performed on the filename used. Enter filename in field provided.

Notify - Enables the test result notification window. This window will be displayed when a test script completes and will show the overall result and play a sound. Use check box to enable.

Interactive Test Order - Using this option, the order of interactive tests in a test list can be set. Use the dropdown to choose Default, First, or Once.

Default – Interactive tests will run in the order that they are listed in the test script.

First – All interactive tests will be run at the start of each lap before all remaining non interactive tests are run.

Once – All interactive tests will be run at the start of the first lap before all remaining non interactive tests. In subsequent laps, these interactive tests will be skipped.

Stop Script - This option will configure the interface's operation once a test script has been run from the command line. Use the dropdown box to configure.

Exit – Interface will exit after script is complete.

Remain Open – Interface will remain open after script is complete

Remain Open on Fail –Interface remains open if failures are detected, if no failures are detected then the interface will close after script is complete.

Predefined Test List - This list of predefined tests scripts can be used in interfaces for a quick access list. Implementation is application specific. This function allows you to add your own test scripts to the interface under diagnostic tools>Select diagnostic tests by script.

UEFI Test Order – This controls how UEFI tests are executed within a test script.

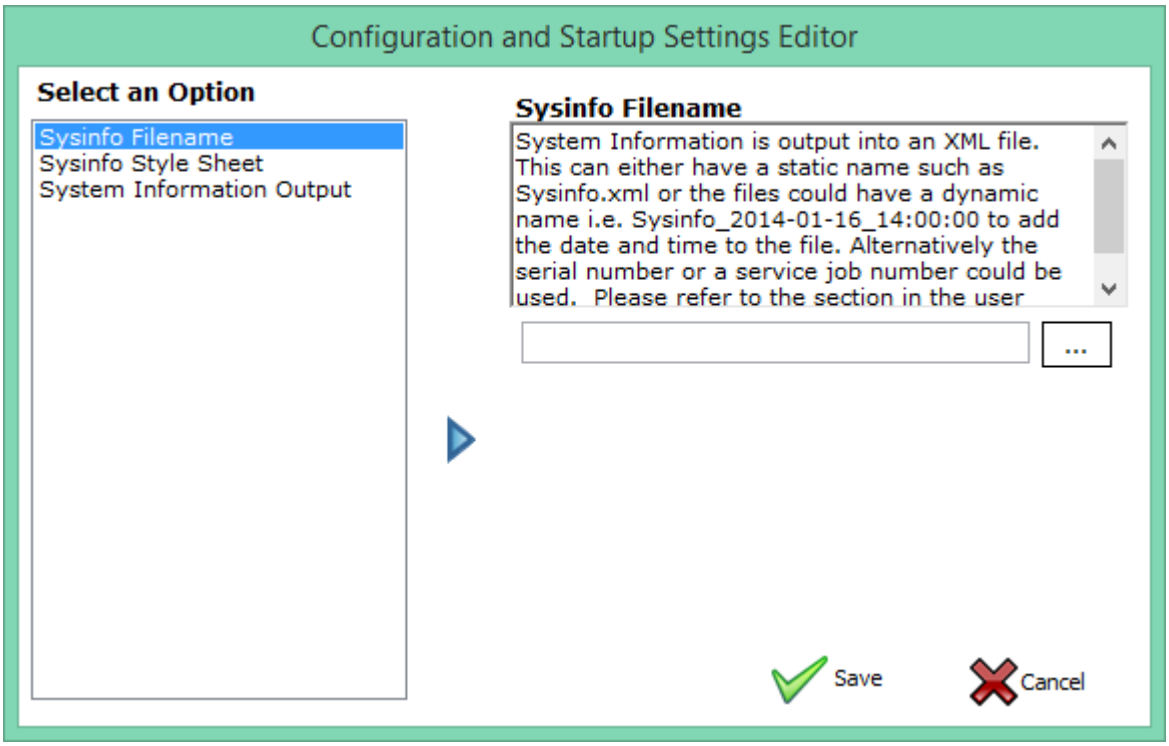
First – All UEFI tests are executed first, if multiple passes are selected in the script then the UEFI tests will be executed at the start of each pass.

First Only – All UEFI tests are executed at the start of the test script, if multiple passes are selected in the script these tests will be skipped in subsequent passes.

Last – All UEFI tests are executed last, if multiple passes are selected in the script these tests will be executed at the end of each pass.

Last Only – All UEFI tests are executed at the end of the script, if multiple passes are selected in the script these tests will only run in the last pass and will be skipped in preceding passes.

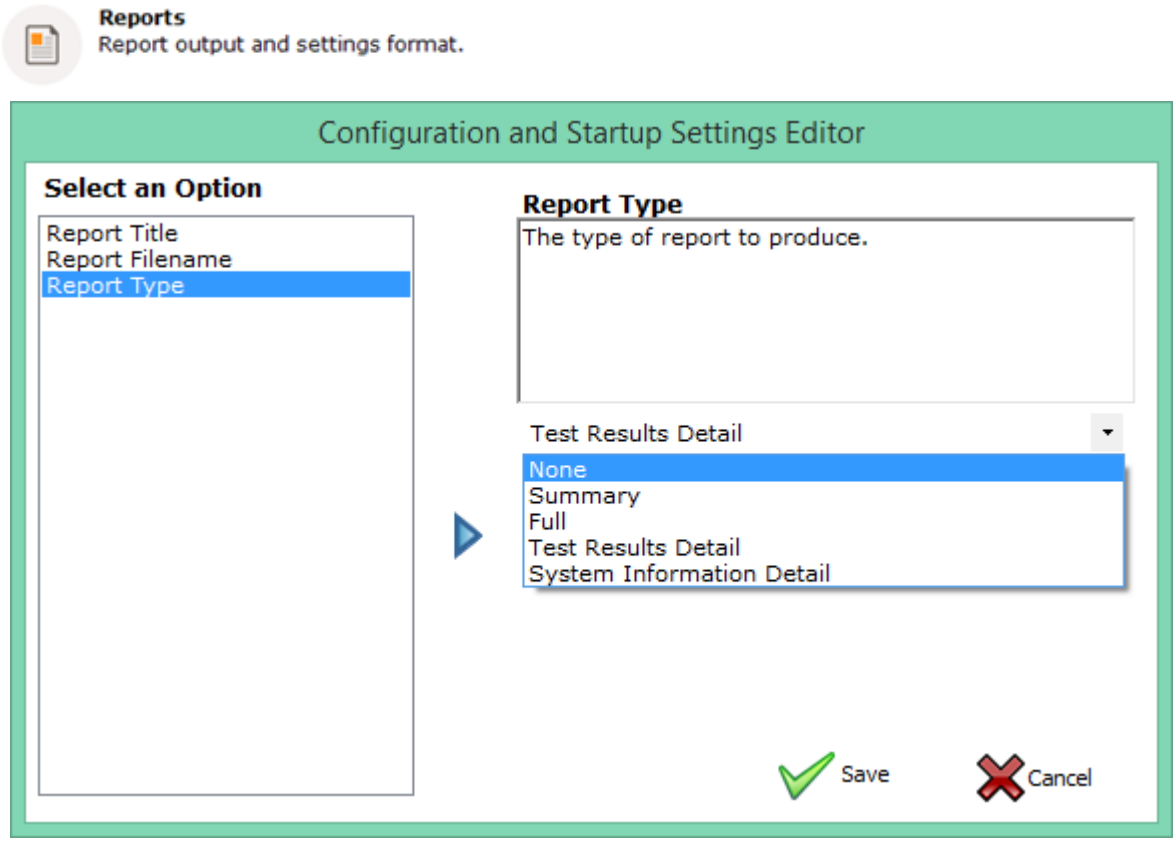
System Information
Format output of system information.



Sysinfo Filename - The filename of the system information XML output file. This must have the .XML extension. This option is only used for the standard Sysinfo output. This option supports user defined fields. Click the button at the right to navigate to the file.

Sysinfo Style Sheet - The style sheet to specify in the top of the Sysinfo xml output. This is only used for the standard Sysinfo output. No checking is performed on the filename used. Enter filename in field provided.

System Information Output – This is a checkbox that when set will enable the system information to be output each time the application is run.

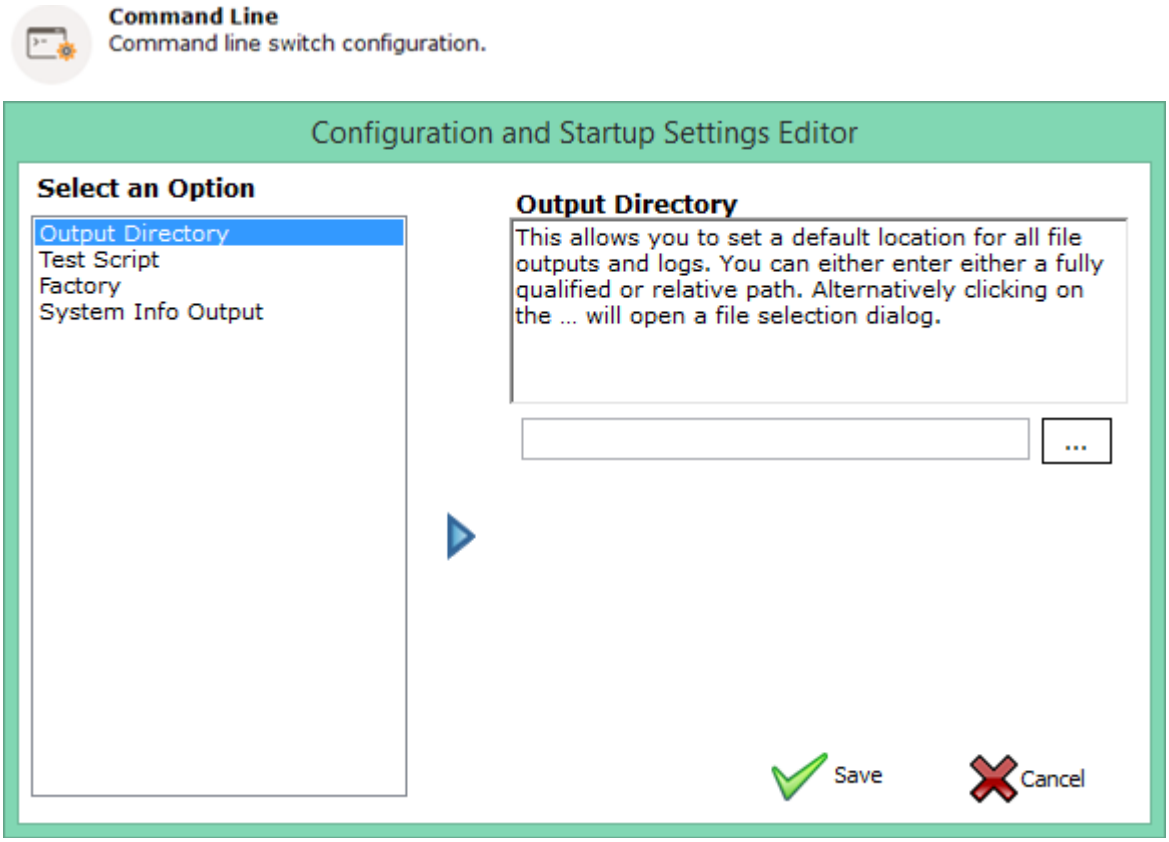


Report Title - The title used for the PDF report. Enter title in field provided.

Report Filename - The filename of the output PDF report. This must have the .PDF extension. This option supports user defined fields. Click the button at the right to navigate to the file.

Report Type - The report can be configured to include varying levels of detail, these are:

None – No report is produced
Summary – A single page summary with overall status is produced.
Full – A report with all test result and system information detail
Test Results Detail – The summary page and the detailed test results
System Information Detail – The summary page, Detailed Test results and system information



Command Line
The Command Line option contains various options to control the behavior at the startup of the program and to input variables.

Output Directory - The output directory for Result and Sysinfo output files. This may be a relative or full file path. Click the button at the right to navigate to the file.

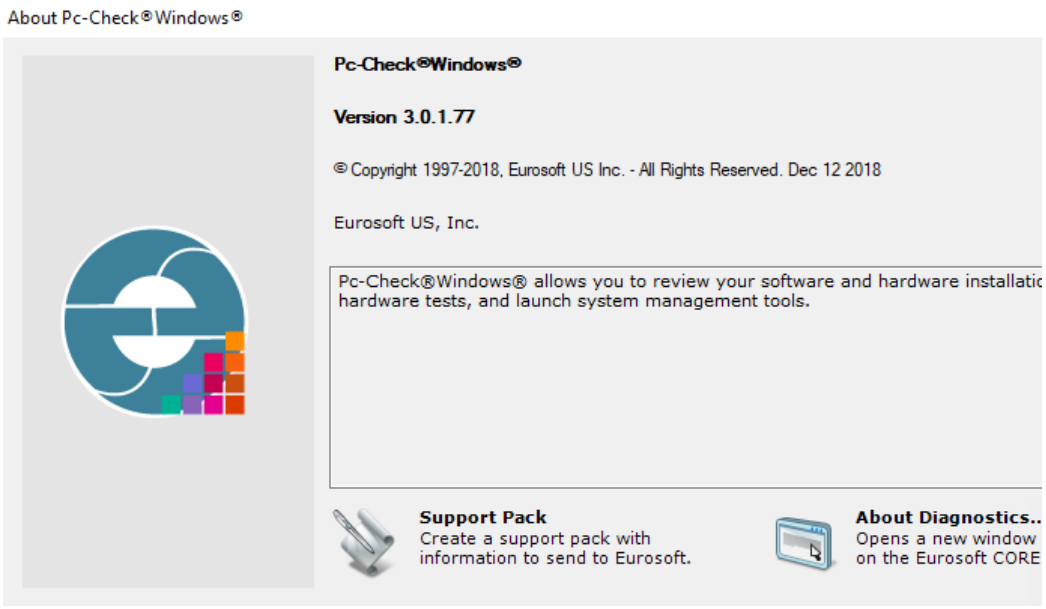
Test Script - The test script to run automatically on start up. By default the application will close once tests are complete. Click the button at the right to navigate to the script file.

Factory - Determines if the tests should run in interactive mode. Use checkbox to enable.

System Info Output - When enabled, the standard system information XML report will be created on shutdown. Use checkbox to enable.

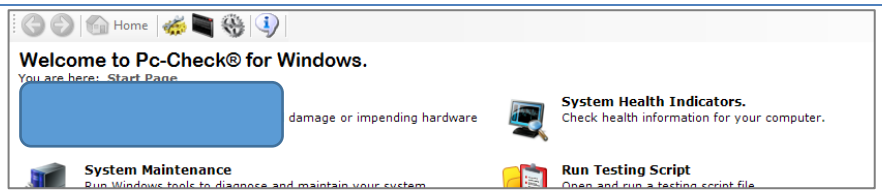
About

The About windows provides information about Pc-Check Windows and provides options to create a Support Pack and open the About Diagnostics window.



Support packs contain all of the files and information that Eurosoft requires when assisting with any issues.

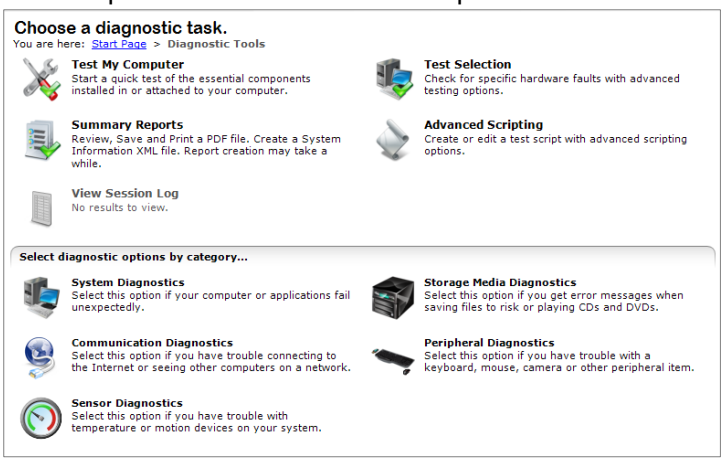
Diagnostic Tools



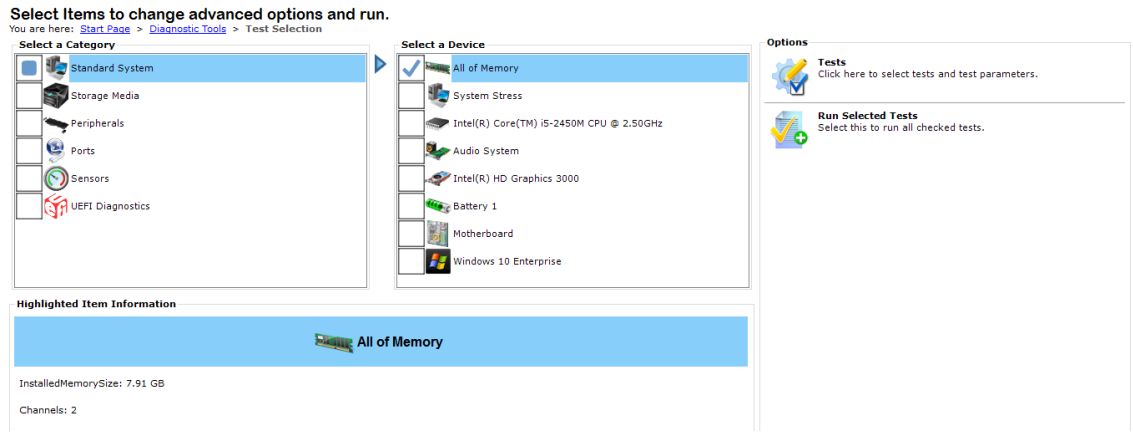
The majority of diagnostic tasks available in Pc-Check Windows are administered through the Diagnostic Tools menu. The Diagnostic Tools menu is accessed from the Home screen as shown here. Diagnostic testing can be quickly and selectively launched. Diagnostic Tools also provides options for saving and loading custom test scripts.

Test My Computer

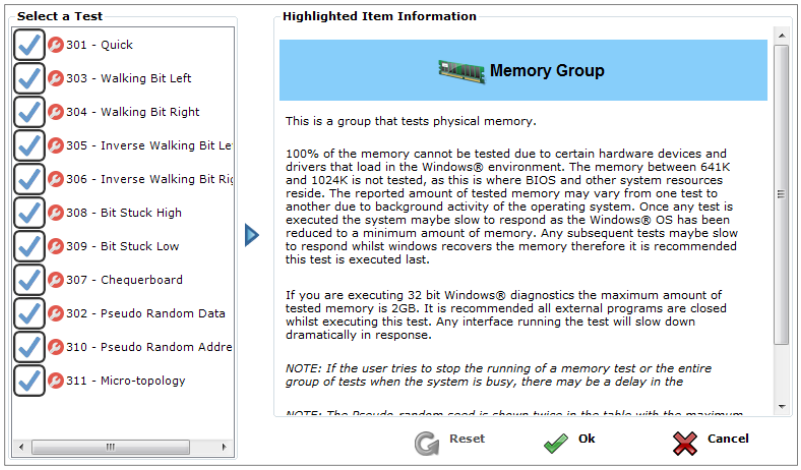
Starts a quick test of the essential components installed or attached to your computer.



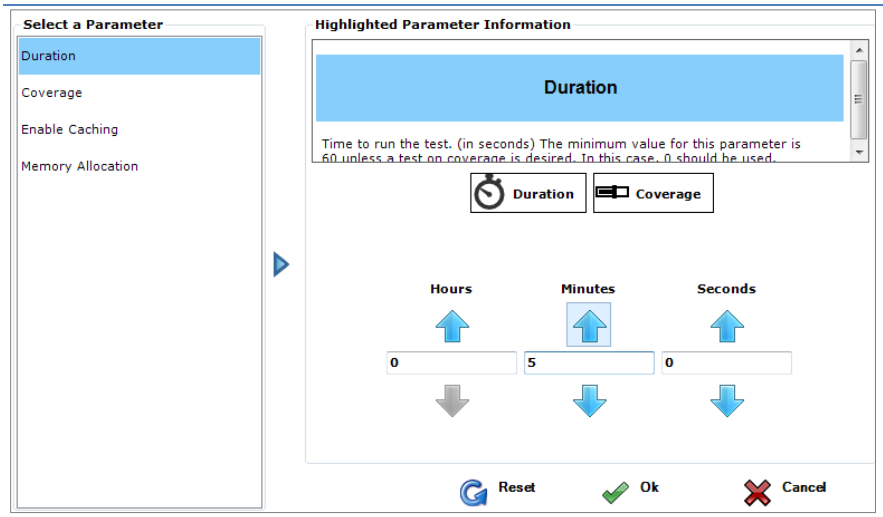
Test Selection



Provides selection and advanced testing options for the immediate launching of tests.

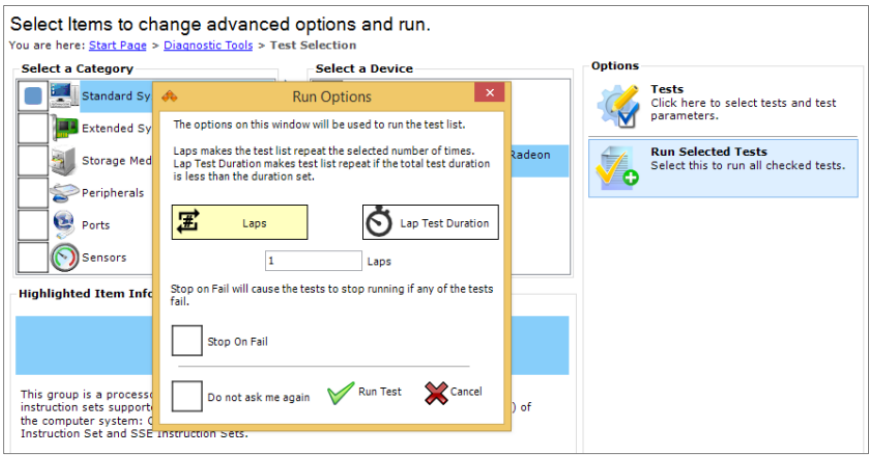


Clicking on a category name will show available devices to test. Clicking a category check box will select all devices in that category with a check in the box next to each device. Clicking on a device will toggle the device test on or off with a check in the box The Tests button opens individual test options.



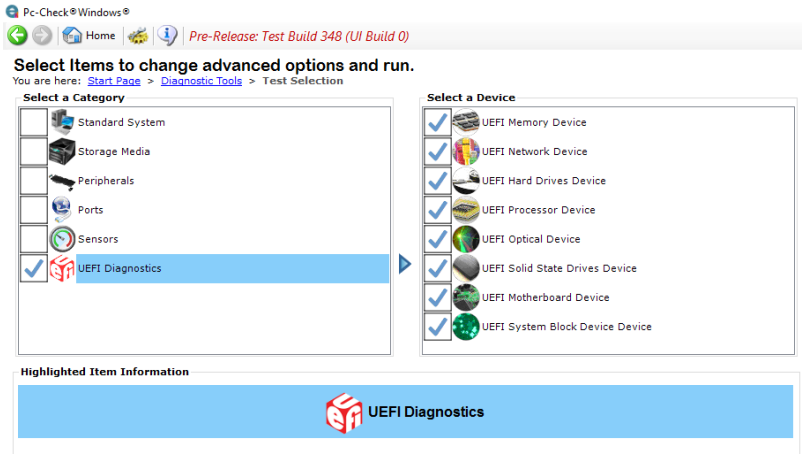
Test parameters are opened through the red circular parameters button located to the right of the listed test
For complete information on tests and testing parameters, please see the Test Descriptions Manual.

When the Run Selected Tests button is clicked, the Run Options pop up appears. Use the Run Options pop up to set the test run to run in laps (iterations), or for a set duration. The Stop On Fail parameter stops the running tests when a failure is encountered.



UEFI Diagnostic Extensions

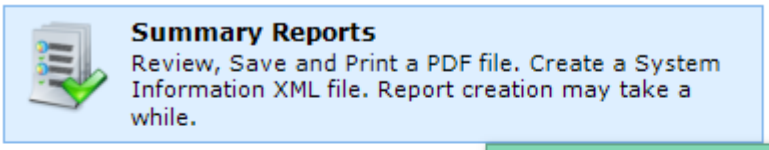
Please contact Eurosoft if you wish to purchase this extension.



This feature allows you to select as part of your test script UEFI diagnostics including tests for:
Memory
Network
Hard drives
Processor
Monitor
Serial Ports
Optical devices
Solid State drives

Motherboard
System block devices
NVMe
For more information view the Pc-Check UEFI test descriptions manual.

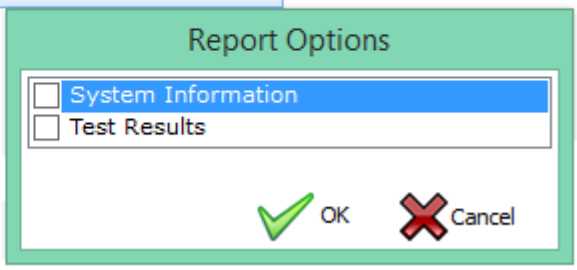
Summary Reports



The Summary Reports option lets you review, save, and print a PDF

Report creation takes a moment to complete.

Report Options let you choose which options you want for your report.



Advanced Scripting

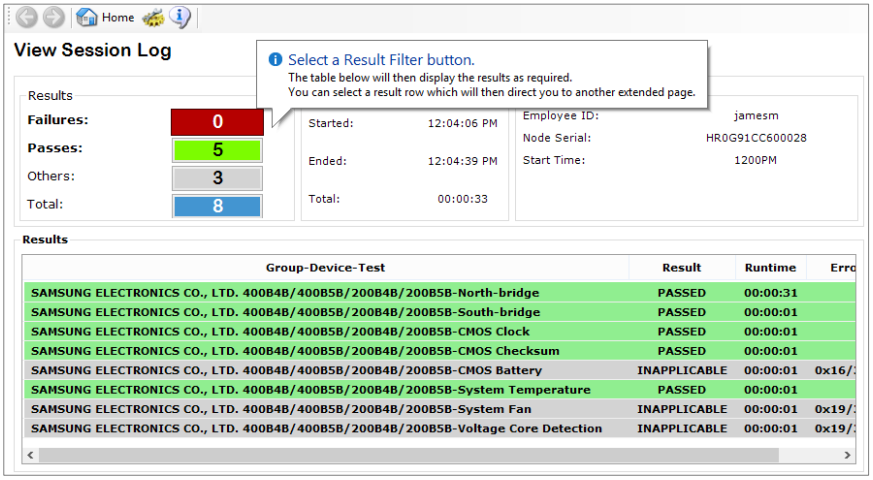
Provides advanced testing options for creating and managing test scripts.

Tests are configured in Advanced Scripting in the same way as they are configured for Test Selection (see Test Selection).

To load or save a test script, use the buttons on the right side of the screen.

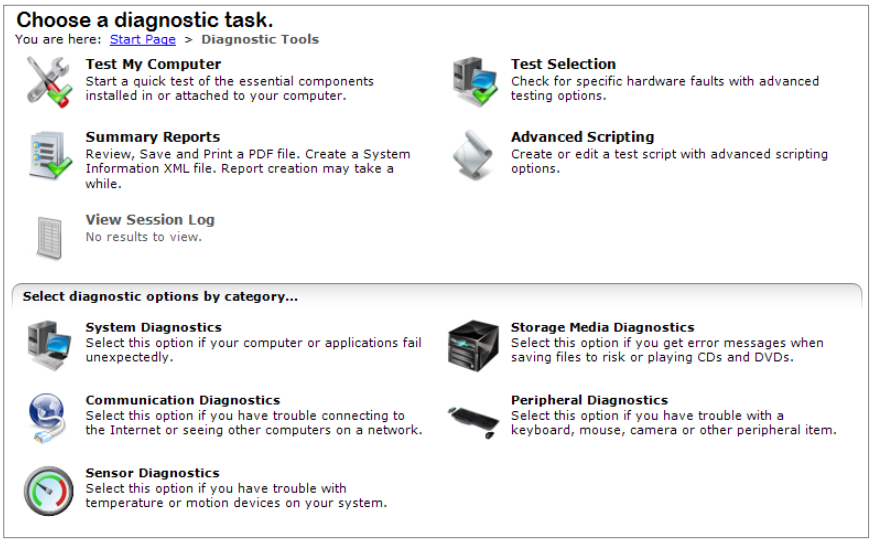
View Session Log

The View Session Log option allows you to view a current log of completed testing activities.



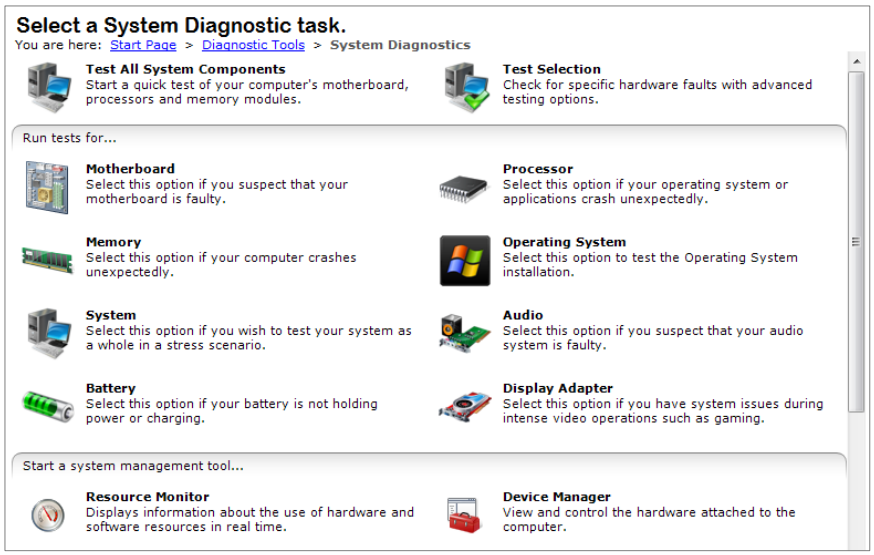
Diagnostic Options by Category

The Select diagnostic options by category menu provides options for executing diagnostic testing on categorized components.



System Diagnostics

The System Diagnostics menu contains options to quickly test all system components and to run Advanced Testing. The menu also contains options for testing essential components such as Motherboards and Processors. The menu also contains shortcuts to useful Windows tools such as Resource Monitor and Device Manager.



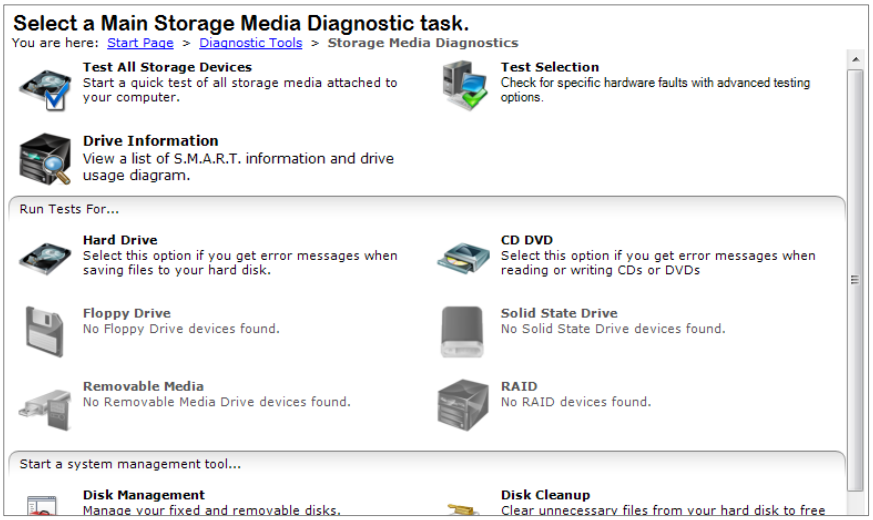
The test groups available through the System Diagnostic Task menu are…

Motherboard
Processor
Memory
Operating System
System
Audio
Battery
Display Adapter

Clicking on any of the above options will initiate testing.

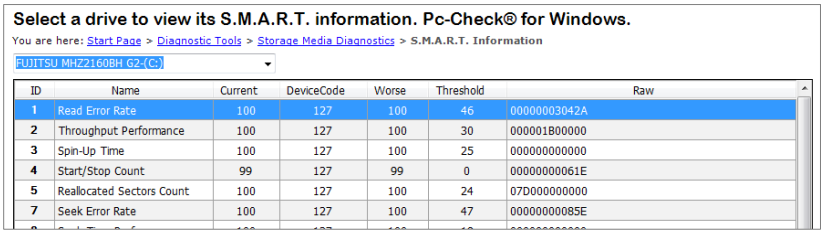
Storage Media Diagnostics

The Storage Media Diagnostics menu provides options to test storage devices. The menu includes an option to test all storage devices as well as an option to initiate Advanced Testing. The menu also includes options to launch useful Windows storage device tools such as Disk Management and Disk Cleanup.



Drive Information

Device Information shows the S.M.A.R.T. log information for the installed hard drives.



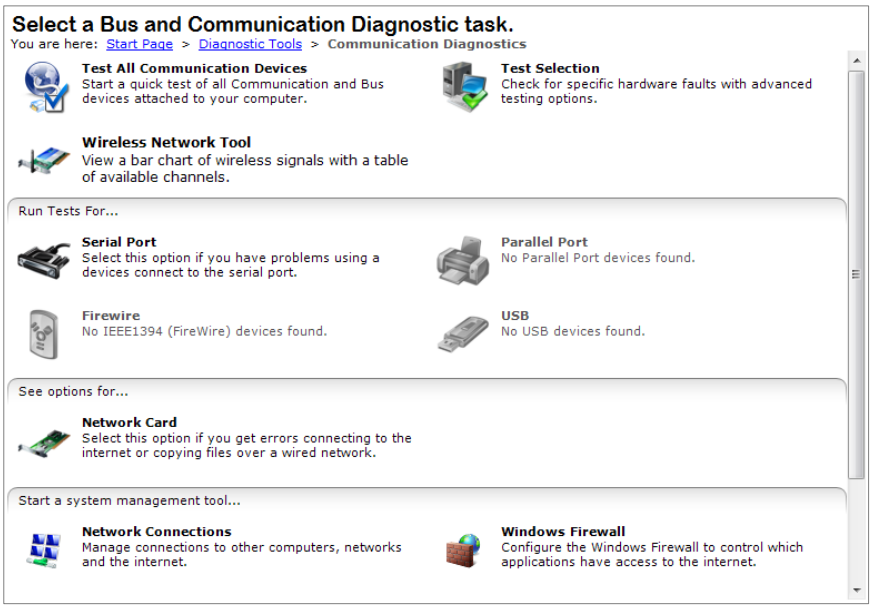
ID	Name	Current	DeviceCode	Worse	Threshold	Raw
1	Read Error Rate	100	127	100	46	00000003042A
2	Throughput Performance	100	127	100	30	000001B00000
3	Spin-Up Time	100	127	100	25	000000000000
4	Start/Stop Count	99	127	99	0	00000000061E
5	Reallocated Sectors Count	100	127	100	24	07D000000000
7	Seek Error Rate	100	127	100	47	00000000085E

The tests available through the Main Storage Media Diagnostic Task menu are…

Hard Drive
CD DVD
Floppy Drive
Solid State Drives
Removable Media
RAID

Clicking on any of the above options will initiate testing.

Communication Diagnostics



Communication Diagnostics menu provides options to test communication components such as network cards and peripheral ports.

The menu provides an option to test all communication and bus devices along with the Advanced Testing option.

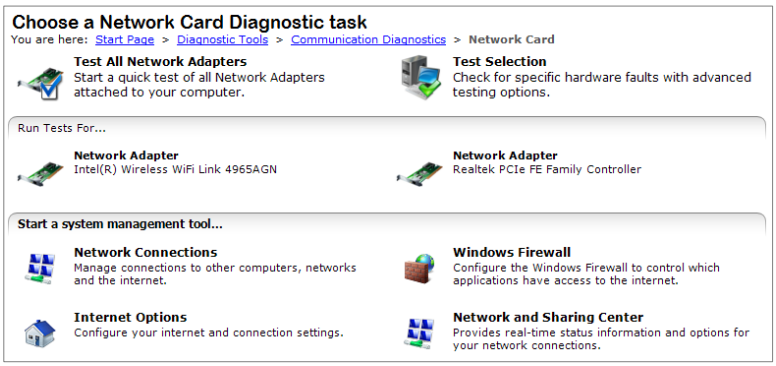
This menu also provides shortcuts to relevant Windows configuration tools such as Network Connections and Windows Firewall.

The tests available through the Bus and Communications Diagnostic Task menu are…

Serial Port
Parallel Port
Firewire
USB
Network Card

Clicking on any of the above options will initiate testing.

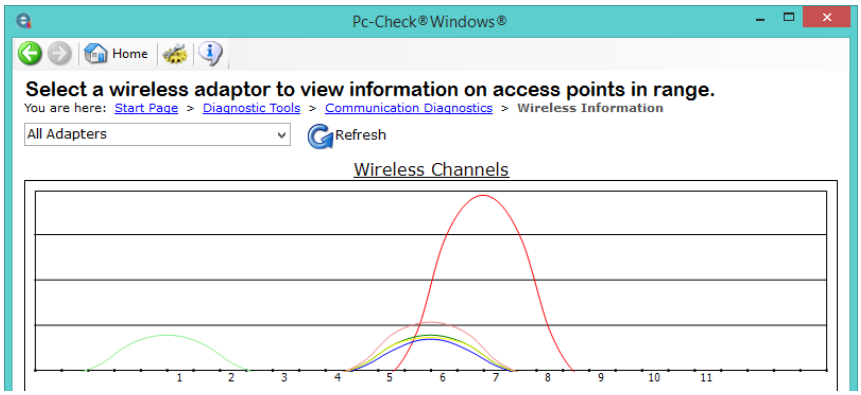
Network Card Diagnostic Task Menu



Test all network adapters, advanced testing, test individual network adapters, and system management tools.

Wireless Network Tool

Use the Wireless Network Tool to use an installed adapter to scan for wireless access points.



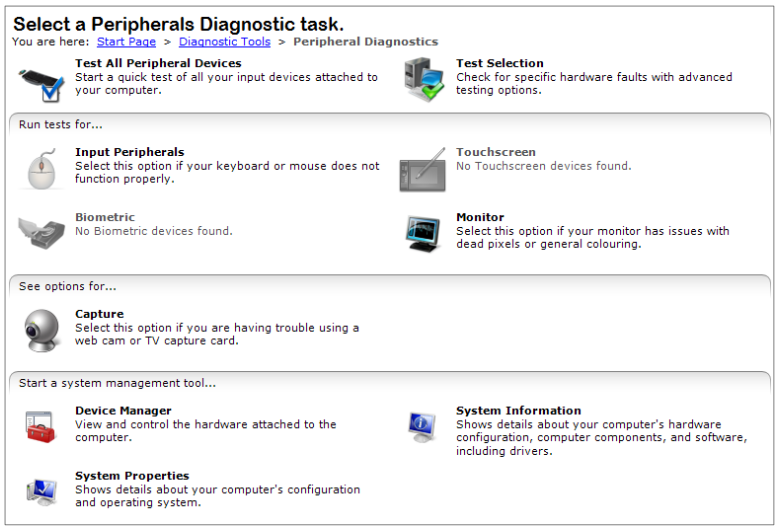
Peripheral Diagnostics

The Peripheral Diagnostics menu provides options to test PC peripherals such as mice, keyboards, and touchscreens. The menu provides an option to test all peripheral devices as well along with the Advanced Testing option.

The menu also provides shortcuts to Windows management tools such as Device Manager and System Properties.

The tests available through the Bus and Communications Diagnostic Task menu are…

Input Peripherals
Touchscreen
Biometric
Monitor
Capture



Clicking on any of the above options will initiate testing.

Capture Diagnostic Task Menu

Test all video capture devices, advanced testing, and test individual capture devices.

Sensor Diagnostics

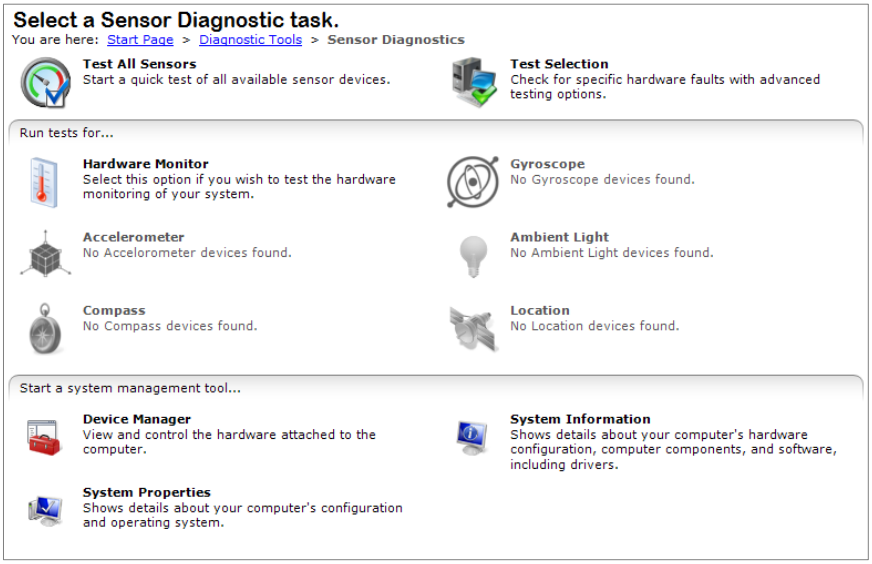
The Sensor Diagnostics menu provides options to test sensors such as temperature sensors, accelerometers, and gyroscopes.

The menu provides an option to test all sensors as well along with the Advanced Testing option.

The menu also provides shortcuts to Windows management tools such as Device Manager and System Properties.

The tests available through the Sensor Diagnostic Task menu are…

Hardware Monitor (sensors)
Gyroscope
Accelerometer
Ambient Light
Compass
Location (GPS)



Command Line Options

You can use command line options with the application using CORE. The following is the list of options available:
Application.exe [/? | /AUTOBOOT | /R <filename>] [switches]

The application available column indicates if it is optional or mandatory Eurosoft CORE implemented. If it is optional please consult your application product as appropriate.

Option	Description	Application Available
/?	Displays on screen a list of command line options that can be used. The Application will close after the options have been shown.	Mandatory on screen list – optional if the application exits
/BLOCK	This option is used to control license BIOS locking. Please contact Eurosoft for more information regarding this feature.	Mandatory
/DF <filename>	Loads a user-specified diagnostic file. This file is used to configure some diagnostic groups, CORE and Application interface. This file must have the ".xml" file extension. This filename can optionally contain a relative or full file path. In terms of configuring the interface, see "Configuration File" for further details. In terms of configuring the groups, please refer to the Test Descriptions document. ***Note:*** *If EDF_CFG.xml is present in the Application directory then this option cannot be used.*	Mandatory
/DIR *<path>*	Re-routes output event and log XML files from the Application directory to another directory. Useful when on read only media. If necessary the directory is created. If files exist these will be overwritten without error.	Mandatory

/ED	Enumeration Debug Log. Eurosoft may request that you run this command line option to gain extra information on initialisation. This will populate the Event log with additional enumeration information and will create the EDF_Trace.log file.	Mandatory
/FACTORY	Runs the tests in non-interactive mode. This mode will not prompt the user for test items such as whether or not loopback plugs are installed.	Mandatory
/OPL	This option is for Eurosoft use only.	Mandatory
/R *<filename>*	Loads and runs the specified script and then shuts down the application. Only groups required to run the script will be loaded. The script must have the .xml extension. This filename can optionally contain a relative or full file path. The shutdown behaviour can be changed using the STOPSCRIPT configuration option. For more information see 'Configuration File' section.	Optional if implemented
/SD	Sensor Debug log. Eurosoft may request that you run this command line option to gain extra information on hardware sensors. This will create the SensorDebug.log file.	Mandatory
/SXML	This option enables the creation of the sysinfo file when the application is shutdown. This file will contain system information from the groups loaded. This option will create the file of the classic log format witch is documented in the appendix. The filename can be configured for this option using the configuration file. For more information see the 'Configuration File' section.	Mandatory-optional at what point it occurs and if it is automatic

Note: *When using the /R option CORE will only load the groups included in the Test script and will close automatically upon test completion.*

Note: *Should you wish to load a group but not run any tests for it, (i.e. to get system information for devices in the group) then simply add the group to the Custom Test Script without any tests associated with it as in the example below.*

```
<?xml version="1.0" encoding="UTF-8"?>
<Script RunType =\"LapCount" RunCount="1" StopOnFail="0" >
    <Group Group="1000" GroupName="Memory">
    </Group>
    <Group Group="5500" GroupName="Processor">
    </Group>
</Script>
```

Application Return Codes

CORE has a number of return codes that can be returned when the application is shut down, designed for use in automated or batch running scenarios. The following is a list of return codes provided by CORE. The first error set will be the return code.

Return Code	Meaning
0	No other return code event has returned.
1	General Failure. CORE has failed to run successfully.
2	Error. CORE has encountered a Major fault. The event log will contain more information on this error.
3	License Failure. CORE failed to run due to a license issue.
4	Test Failure. One or more of the tests that ran failed.
6	Eurosoft use only.

NOTE: *The support of these application return codes is Application Optional.*

Interactive Test Panes

Most Interactive Tests will create an Interactive Test Pane when the test is started. You can use the command line option /FACTORY to run the application and its tests in non-interactive mode.
When this option is set, the interactive tests are skipped, returning the Not Available result. It is a convenient way to run a test script that contains interactive tests without requiring input from the user. In this mode all test accessories must be inserted. If the test accessory is not inserted the result will be FAIL or NOT AVAILABLE dependent on the test. The result codes are explained in greater detail in the test descriptions document.
Messages and prompts are also suppressed or added with timeouts to ensure that CORE can be run in an automated fashion.

Note: *The NOTIFY configuration option does not have any timeout and will remain until user interaction occurs to ensure the overall result is seen. The notify window must be closed before the application will close when running using the /R option.*

Result Log

This will log information on your testing session in the log file, named by default as <AppTitle>_Results.xml. Where the <AppTitle> is application specific.

The log file contains the following information:

- Test Summary - a list of the modules tested and whether the module passed or failed. This does not include individual tests run. If one test in a module fails, the report lists failed for the entire module.
- Date and time the test started or completed
- Group number
- Test number
- Device number
- Test started
- Pass or fail information
- Additional test information, such as retries.

For more information on the XML format can be found in the section 'Results XML Log'. The log file is created when the first test is started.

Note: *The results log, /LOG and /SXML filenames can be dynamically configured using the configuration file. Information on dynamic filenames can be found in the 'Dynamic Fields' section.*

Error Codes and Extra Information

This section gives detailed information on the error code and extra information associated with this code. This currently applies to the results XML log description string and the error code and extra information with the status results window.

An example XML log entry is defined below for explanation.

```
<LogEvent datetime="2014-05-30T13:43:17+01:00" group="5611" test="302" device="1" message="7"
lapcount="1" pferror="0000000000000000" pffile="" groupname="Audio Group" testname="Audio Connection"
devicename="Audio System" description="NOT AVAILABLE - 0x10/014-0081 - 0008000000020002"/>
```

The general format of an error code is defined below:

0x<diagnostic group error code>/<extra information code>-<diagnostic error instance>

i.e 0x10/014-0081 would state

0x10	The count of jacks did not change
014	Detected jacks before and after testing (AUDIO specific). All of the diagnostic extra information codes are documented at the front of the test descriptions document.
0081	The error instance specifically within the group, in this case this is required if the error occurs multiple times. This is useful for Eurosoft in order to better support a test not passing.

The extra information is split into 2 sections the top 16 bits is the type of extra information and the bottom 48 bits is the actual extra information itself.

No extra information	0000
Coverage in KB	0001
Coverage in MB	0002
Coverage in percent	0004
Diagnostic Group Error	0008
Diagnostic Temperature Error	0010

i.e 0008000000020002 would state

0008	Diagnostic Group Error
000000020002	2 jacks were found before and after testing

Note: *In the case of a diagnostic group error (as above) the extra information is entirely diagnostic group and error specific and has no fixed format available. Assistance should be sought from Eurosoft if the extra information is not immediately evident using the extra information code.*

System Information Log

If the /SXML option has been used then the system information is output into the Sysinfo.xml file. The format of this file can be found in the 'XML Specification' section. Further information can be found in the 'Command Line options' section.

Note: *The /SXML filename can be dynamically configured using the configuration file. Information on dynamic filenames can be found in the 'Dynamic Fields' section.*

Icon Customisation
The images used for the icons and buttons within CORE can be customised as required. The image files used for CORE can be found in the Interface\Icons directory.

Note: *The image format and dimensions must be kept the same when replacing the existing images in these directories.*

Package Customisation

Reducing Execution time:

The CORE execution time can be reduced by removing unwanted diagnostic groups. E.g. if the parallel port group is not required 20X.dll can be removed. The more groups that are removed the quicker the time will be to execute the application and the less information is gathered for any system information reports.

Reducing the size of the package:

By removing groups this also has the advantage of reducing the size of the package for copying either over a network or onto media which has a reduced size. If a group is not required the xtra*<group number>* should be removed as well as the root group component file.
It is possible by removing the relevant platform directories and file content to produce only the relevant platform required to be supported.

Note*: It is recommended to back-up your application before customising your package for testing. Further details maybe requested from Eurosoft.*

Components that can be removed:

File	Customise	Comment
PDF.dll	No	PDF Functionality removed. This includes the user guide and test description information on the main interface.
EDF\<platform>\<Group>	No	Removes functionality on groups.
EDF\Help\Test_Descriptions.pdf	Yes	Consult Eurosoft to create languages.
EDF\Help*.xml	Yes	Consult Eurosoft to create languages.
EDF\Xtra\<file>	No	Removes certain diagnostic group functionality. All components can be removed if necessary. Please consult Eurosoft for more assistance.
Interface\Notify<result>.jpg/.wav	Yes	Sound and image files for the NOTIFY configuration option.

Configuration File XML Specification

Eurosoft CORE reads entities from the specified configuration file for determining how some features of the interface will operate. These entities are automatically updated depending on how CORE is run. When any tag value is empty then the option is considered not used.

Any options that can be configured by both the command line options and the configuration file must be removed from the configuration file to be used by the command line. When using the configuration file in this way it is recommended to mark the file as read only or to use the command line option /LCFG to ensure that the missing tags are not re-inserted into the configuration file.

As per the command line options each control is marked if it is optional or mandatory implemented within Eurosoft CORE.

Eurosoft_<id>
This is the tag that CORE reads, where the number should be the Application ID ranging from 1 to 15. The sub entities of this tag should contain the required settings.

DIR
The DIR tag operates the same as the /DIR command line option. This tag will take priority over the options entered on the command line. This tag should be deleted if the command line is used.

Feature: Application Mandatory

FACTORY
The FACTORY tag operates the same as the /FACTORY command line option. This tag will take priority over the options entered on the command line. This tag should be deleted if the command line is used.

Feature: Application Mandatory

FactoryTestOrder
The FactoryTestOrder tag controls the automatic re-ordering of interactive tests from any test script or selection run. This tag must be one of the following options.

Option	Description
0	Default. No re-ordering will occur.
1	All interactive tests will be run at the start of each lap before all remaining non interactive tests are run.
2	All interactive tests will be run at the start of the first lap before all remaining non interactive tests. In subsequent laps, these interactive tests will be skipped.

Feature: Application Mandatory

FilenameResults

This tag defines the filename for the results XML log file used in the section 'Result Log'. This filename must be relative and be of the .xml file type. The filename may be pure text or may use the dynamic fields specified in the 'Dynamic Fields' section.

Feature: Application Mandatory

FilenameSysinfo
This tag defines the filename for the Sysinfo.xml XML log file used in the section 'System Information Log'. This filename must be relative and be of the .xml file type. The filename may be pure text or may use the dynamic fields specified in the 'Dynamic Fields' section.

Note: *The option SXML 1 must be used for the system information log to be created.*

Feature: Application Mandatory

Home
This element is used to create links to scripts on your system. These are shown as a list in the Homepage window of the interface. It contains the Script tag which is explained below.

Script
This tag contains the list of scripts to be used by the application to list pre-configured test scripts and has three main attributes the Path, Description and Title.

The Path attribute specifies the script file. If a script file is specified, then the file must exist on your system. A relative or full path can be used with scripts.

The Title must exist, while the Time and Description attributes are optional and may be used for test scripts to specify an estimate for the runtime.

```
<Home>
        <Script Path="ScriptQ.xml" Title="Quick test script" Description="The standard quick test script."
Time="10 Minutes"></Script>
</Home>
```

Feature: Application Optional

InteractiveTimeout

In the case where the Application is in interactive mode, the InteractiveTimeout tag allows a timeout for all interactive tests and windows to be configured. If the duration of an interactive test or message exceeds the timeout time, the interface and diagnostics will change to non-interactive mode. This timeout is in minutes.
To return to interactive mode after the timeout occurs, the application will need to be restarted.
If this is not set or no configuration file is used, the timeout is set to an hour. A value of 0 will disable the timeout entirely.

Feature: Application Optional outside of testing.

LIC
Please contact Eurosoft for more information regarding this feature.

Feature: Application Mandatory
LOG
Sets the filename of the Report to be created when the application closes. The report file created will be a Full Report with Test Results and Device information.
This filename must be relative and be of the .pdf file type. The filename may be pure text or may use the dynamic fields specified in the 'Dynamic Fields' section.

Feature: Application Mandatory

NOTIFY

This option enables the Test Result notification window and sound when any series of tests complete. The sound output entirely relies on system configuration and capability of operation. We recommend that audio loopbacks are unplugged before the tests complete.
The window and sound remain until there is user interaction.
The window image and sound can be customised by modifying the 'Notify<result>' files in the Interface directory.

Feature: Application Mandatory

RFC1766

This element contains the entities that specify the license file list used to select the license file to use dependent on the system locale.

RFCENTRY

The RFCENTRY tag contains the RFC1766 language id and the license file path to use on this language. This tag defines the license file to use for each Language ID. If no match is found for the current language id, then the default license file is used either defined in the LIC attribute value or default within the product.

```
<RFC1766>
    <RFCENTRY Lang="en-gb" File="License_en-gb.lf"></RFCENTRY>
    <RFCENTRY Lang="zh-cn" File="License_zh-cn.lf"></RFCENTRY>
</RFC1766>
```

Feature: Application Mandatory

RXSL

This option chooses the XSL style sheet that should be used with the results file. This will add the xml-stylesheet tag to the top of the file referencing the chosen style sheet.
NOTE: *There is no validation performed on the XSL file used or the filename itself.*

Feature: Application Mandatory

STOPSCRIPT

This tag controls stopping after a script is set on the command line using /R. The stopscript option should be one of the following options:

Option	Description
0	The application will shut down after testing is complete
1	The application will not shut down after the tests are complete.
2	The application will not shut down only if a test fails.

Feature: Application Optional

SUPPORT

The option enables the creation of a Support file pack when the application is shutdown. This file can be sent to Eurosoft for assistance if required. Manual support file creation is application specific.

Feature: Application Mandatory

SXML

This option enables the creation of the sysinfo file when the application is shutdown. This file will contain system information from the groups loaded.
This tag should be deleted if the command line is used.
If the option is 1, the classic log format is produced which is documented in the appendix. The filename can be configured for this option using the FilenameSysinfo option.
The following additional options are available:
A PCA (Preboot Pc-Check) Combined Log, with an option of 2, 4 or 5.
A PCA Log with no Pc-Check log files, with an option of 3.
Please contact Eurosoft for assistance with using these additional options.

Note: *If the SXML mode used on the command line is different to the one used in the configuration file, the configuration file will be rejected.*

Feature: Application Mandatory

SYS_MEDIA_DIR

This tag is used in conjunction with /SXML 2, 3 and 4 as the output directory on the removable media.

Feature: Application Mandatory

SXSL

This option chooses the XSL style sheet that should be used with the Sysinfo file. This will add the xml-stylesheet tag to the top of the file referencing the chosen style sheet. This is only used with the SXML 1 configuration file option.
NOTE: *There is no validation performed on the XSL file used or the filename itself.*
Feature: Application Mandatory

Title

This tag defines the Title used across the product. This will change the name of the windows displayed by the interface.

Feature: Application Optional

TPF

This tag defines the TPF (Technician Perspicuity Factor) to use for the interface. This setting is used to control the workflow of the technician where the lower the setting, the more opportunity for exploratory testing outside of a fixed workflow is available. The level indicates the quality of insight and judgement of the operator. The setting must be one of the following:

Value	Meaning
0	No functionality is restricted and all advanced features are available. This level is most appropriate for operators who are configuring test scripts for basic technicians and advanced technicians who have no restriction on their workflow.
1	Intermediate features are available such as basic log information. Advanced features such as script creation may be unavailable. This level is most appropriate for operators who are allowed to interpret a result from an individual test combined with any system information available.
2	Only basic features are available. This level is most appropriate for those operators who only view the results on the interface and perform a fixed workflow depending on that result.

NOTE: *Actual functionality available is application specific, the user should consult the application as appropriate.*

Feature: Application Optional

UserDefinedFields

This element contains the User Defined Fields specified in the 'Dynamic Fields' section. The UserDefinedFields tag also configures the title that will be used for the user defined fields entry box.

UserDefinedField

This tag is used to define a user defined field.
The User Defined field will only be valid if it contains a valid ID and Name. The Name attribute will be used as the label for the entry box on the interface and must be less than 16 characters. The ID attribute must be between 0 and 5.

The optional Value attribute sets the start value of the user defined field. This attribute can use the defined fields in the section 'Defined Fields'.

The optional Mask attribute sets the mask to use to validate the user defined field. If a field is found to be invalid then the user will be prompted to enter a valid value from the start-up dialog. The format of the mask can be found in the 'User Defined Field' section.

The optional MaskExample attribute allows an example of a valid value to be entered to show the user what is expected. This attribute value must pass validation against the mask to be used.

The Locked attribute, when set to 1, will stop the user from being able to change the value from its start value. If a mask is set and the value fails validation, the user defined field will only be editable from the start-up dialog.

Feature: Application Optional in terms of the interface displaying and requesting parameters. It is mandatory for the output report.

```
<UserDefinedFields Title="User Params">
        <UserDefinedField ID="0" Name="Tester" Value=":W1:" MaskExample="" Mask="" Locked="1"/>
        <UserDefinedField ID="1" Name="Time" Value=":QHM:" MaskExample="" Mask="" Locked="1"/>
        <UserDefinedField ID="2" Name="OS" Value=":EOS:" MaskExample="" Mask="" Locked="0"/>
        <UserDefinedField ID="3" Name="Order Number" Value="12345-12347" MaskExample="11111-
11111-11111" Mask="NNNNNONNNNNonnnnn" Locked="0"/>
        <UserDefinedField ID="4" Name="Memory Size" Value=":G100X1:" MaskExample="" Mask=""
Locked="0"/>
        <UserDefinedField ID="5" Name="MAC Address" Value=":G221X5242885:"
MaskExample="A1:B2:C3:D4:E5:F6" Mask="XX#XX#XX#XX#XX#XX" Locked="1"/>
</UserDefinedFields>
```

User Defined Fields and Dynamic Filenames

User Defined Fields are fully customisable fields that may be shown within the application and will be written at the top of the standard report. These may also be shown in the Start-up Dialog if user interaction is required. The application will need to be consulted on operation as required.

The user defined field syntax is used to create Dynamic Filenames. The User Defined Fields are specified in the configuration file.

The user defined fields provide a way to add additional information from a variety of sources such as user input, device attributes and environment variables. These fields can be validated and used to customise the standard report and output filenames.

Dynamic Filenames are customisable filenames for the Report, Results log and System information log.
These fields change depending on user input and system information to allow for machine dependent information to be used. For example, a user defined field can be set to contain the amount of system memory, to display in the top of the PDF report.

The user defined field syntax is specified using colon characters, and are listed below. Some fields cannot be used within the User Defined Fields configuration and are marked in the table.

Field Name	Field Format	User Defined Field Compatible
Environment Variable	:E<Variable Name>:	Yes
Group Attribute	:G<GroupID><AttributeID>:	Yes
Application Start	:Q<Format>:	Yes
Report Time	:R<Format>:	No
Special	:S<FieldID>:	No
User Defined	:D<FieldID>:	No
Windows	:W<FieldID>:	Yes

Note: *If any errors are encountered while processing the dynamic fields the errors will be logged in the event log.*

Environment Variable Field

The Environment variable field is used to specify a Windows® environment variable that the application will read and use. The Environment variable is read once at start-up only.
For example, to use the environment variable OS, the field :EOS: should be entered into the configuration file. Where used on a Windows® NT machine, this will automatically be populated with the value "Windows_NT"

Group Attribute Field

The Group attribute field is used to specify an attribute that will be used. CORE will gather the group information for the specified group at start-up in order to retrieve the attribute. If the attribute cannot be found for device 1, then the text NULL will be used instead. This exclusion also includes using an automatic test script to remove the group being loaded.
The group ID used for this field must use the platform independent ID as used in the Test Descriptions manual.
For example, to use attribute 1 of the 1000 Memory group, the field :G100X1: should be entered into the configuration file. This will automatically gather group 1000's system information on initialisation and will be populated with the value for attribute 1. For example on a machine with 1gb of system memory this will be populated with "1 gb".

Start Field

The Start field is the time that CORE was started. This field contains a formatter that specifies how the time will be used. The formatter is made of any combination of the following characters.

Character	Description
D	Integer day of the month
M	Integer month
Y	Last 2 digits of year
Y	Year in full
H	Hour in 24 hour format
H	Hour in 12 hour format
M	Minute
S	Second
P	AM/PM
U	Week number
A	Weekday name
B	Month name

NOTE: *Any characters used for the formatter that are not in the table will be ignored.*

NOTE: *To add characters between different parts of a date, multiple fields will need to be defined.*

For example, to get the data in YYYY-MM-DD format, the fields would need to be specified as ":QY:-:Qm:-:Qd:". This will give "2014-10-10" on the 10th of October, 2014.

Report Time Field

The report time field is the time that the PDF report specified with the LOG option is created. If the report has not yet been created, the date and time will be zeroed according to the formatter.
The formatter is in the same format as for the Start field, please refer to the Start field section for an example.

Special Field

The field specifies a type that may change during the running of the Application. The ID must be from the table below.

ID	Description
1	Title. This is the same title used on the interface if the title has been customised.
2	Overall test result – This is the overall test result that will be shown on the Service style report.

For example, to use the overall test result, the field :S2: should be used. If the test has not run, this will be populated with "NotRun", and will be updated as the overall test result changes.

User Defined Field

The ID must be a number between 0 and 5, specifying the field to use.

The start value specified in the configuration file for a user defined field may use the Environment, Group, Start time and Windows fields. If the start value begins with a question mark '?' character then the start-up entry window will be shown to allow user entry at start-up. i.e "?:G100X1"

The user defined fields can be configured with an optional mask. The mask will be used to validate the start value and any invalid fields will need to be corrected within the start-up dialog. The mask length will limit the length of text that can be entered for the field's value and must be made up of any combination of the following characters.

Character	Meaning
N	Any number
N	Any optional number
A	Any letter
A	Any optional letter
X	Any letter or number
X	Any optional letter or number
O	Any character

O	Any optional character
#	Any optional character. This character will be excluded from the filename output.

For more information on configuring User Defined Fields please refer to the 'Configuration File' section.
For example, to use the first user defined field for a filename, the field :D0: should be used. This will contain the same value as entered and shown in the test configuration and top of the standard report.

Windows Field

This field specifies a Windows® specific value that is retrieved at start-up. The ID must be from the table below.

ID	Description
1	Current user Username
2	Machine name

For example, to use the machine name, the field :W2: should be used. This will be populated with the machine name. For example, with a machine named "EurosoftMachine1" then the value will be populated with "EurosoftMachine1" too.

App defined fields

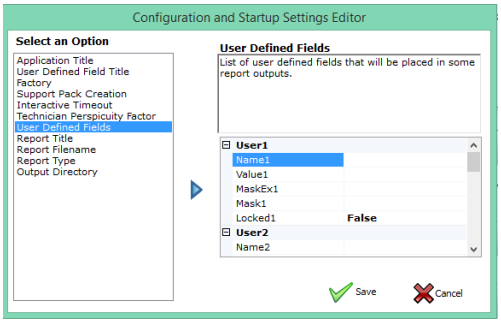
These fields specify additional functions that can be applied to the pdf report output file.

ID	Description
Report_PrefixPages	Add a customised PDF to the standard PDF report as a prefix. File should be named Report_prefix.pdf and reside in the Interface directory.
Report_SuffixPages	Add a customised PDF to the standard PDF report as a suffix. File should be named Suffix_prefix.pdf and reside in the Interface directory.
Report_PrefixFooter	Set to 1 or 0 to turn on or off the inclusion of the footer on your Prefix footer.
Report_SuffixFooter	Set to 1 or 0 to turn on or off the inclusion of the footer on your Suffix footer.
Report_Header_Image	Add a company logo to the report as a header. Recommend the file be 595 by 35 pixels. File should be named Report_Header.jpg and reside in the Interface directory.
Report_Include_Os	Set to 1 or 0 to turn on or off the inclusion of the operating system group information from group 681X such as Installed programs, Installed drivers and OS details.
Report_OrderByResult	Set to 1 or 0 to turn on or off the sorting of the results, firstly Failures, Secondly Pass, Lastly Not Available.
Report_Include_Fail_Trouble	Set to a 1 or 0 to turn on or off the inclusion of failure troubleshooting details in the report.
Report_Include_NotAvail_Trouble	Set to a 1 or 0 to turn on or off the inclusion of N/A not available troubleshooting details in the report.
Report_Colour_PageText	Default colour is black for the page text.
Report_Colour_Borders	Default colour is black for the cell borders.
Report_Colour_DefaultCell	Default colour is white for these description cells.
Report_Colour_DefaultCellText	Default colour is black text for these description cells.
Report_Colour_TitleCell	Default colour is grey for these label cells.
Report_Colour_TitleCellText	Default colour is black text for these label cells.
Report_Colour_HeadingCell	Default colour is black for the sub heading background.
Report_Colour_HeadingCellText	Default colour is white text for the sub title heading text.
Report_Colour_HeadingPage	Default colour is black for the main title heading background.
Report_Colour_HeadingPageText	Default colour is white text for the main title heading text.

Appendix A

UserDefinedFields

User Defined Fields and Dynamic Filenames User Defined Fields are fully customizable fields that may be shown within the application and will be written at the top of the standard report. These may also be shown in the Start-up Dialog if user interaction is required. The application will need to be consulted on operation as required. The user defined field syntax is used to create Dynamic Filenames. The User Defined Fields are specified in the configuration file. The user defined fields provide a way to add additional information from a variety of sources such as user input, device attributes and environment variables. These fields can be validated and used to customize the standard report and output filenames. Dynamic Filenames are customizable filenames for the Report, Results log and System information log. These fields change depending on user input and system information to allow for machine dependent information to be used. For example, a user defined field can be set to contain the amount of system memory, to display in the top of the PDF report.



The Name attribute will be used as the label for the entry box on the interface and must be less than 16 characters. If the start value begins with a question mark '?' character then the start-up entry window will be shown to allow user entry at start-up. i.e "?:G100X1:" The user defined fields can be configured with an optional mask. The mask will be used to validate the start value and any invalid fields will need to be corrected within the start-up dialog. The mask length will limit the length of text that can be entered for the field's value and must be made up of any combination of the following characters.

Character	Meaning
N	Any Number
A	Any Letter
X	Any Letter or Number
O	Any Character
#	Any Character, Character will be excluded from filenames
n,a,x,o	Any optional equivalent to the above.

Environment Variable

Environment variables can also be used. The Environment variable is read once at start-up only. For example, to use the environment variable OS, the field :EOS: should be entered into the configuration file. Where used on a Windows® NT machine, this will automatically be populated with the value "Windows_NT"

Diagnostic Variables

To use values obtained from the diagnostics, that is gathered as part of the system information
The Group attribute field is used to specify an attribute that will be used. The Diagnostics will gather the group information for the specified group at start-up in order to retrieve the attribute. If the attribute cannot be found for device 1, then the text NULL will be used instead.

For example, to use attribute 1 of the 1000 Memory group, the field: G100X1: should be entered. This will automatically gather group 1000's system information on initialization and will be populated with the value for attribute 1. For example on a machine with 1 GB of system memory this will be populated with "1 GB".

Application Start

The time that Pc-Check Windows was started. This field contains a formatter that specifies how the time will be used. The formatter is made of any combination of the following characters.

Character	Description
D	Integer day of the month
M	Integer month
Y	Last 2 digits of year
Y	Year in full
H	Hour in 24 hour format
h	Hour in 12 hour format
m	Minute
s	Second
P	AM/PM
U	Week number
A	Weekday name
B	Month name

NOTE: Any characters used for the formatter that are not in the table will be ignored. NOTE: To add characters between different parts of a date, multiple fields will need to be defined. For example, to get the data in YYYY-MM-DD format, the fields would need to be specified as ":QY:-:Qm:-:Qd:". This will give "2014-10-10" on the 10th of October, 2014.

Windows Field

This field specifies a Windows® specific value that is retrieved at start-up. The ID must be from the table below.

ID	Description
1	Username Current user
2	Machine name

For example, to use the machine name, the field :W2: should be used. This will be populated with the machine name. For example, with a machine named "EurosoftMachine1" then the value will be populated with "EurosoftMachine1" too.

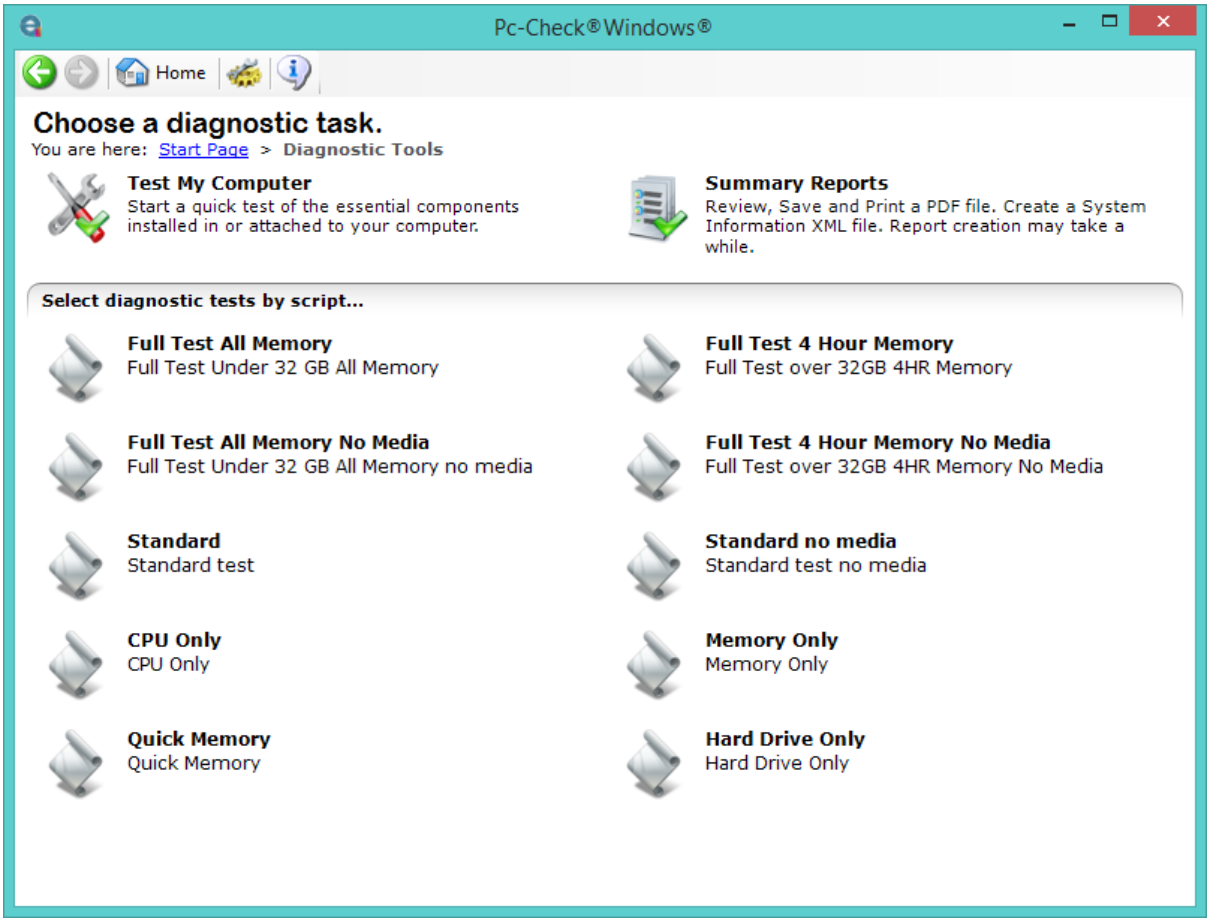
Appendix B

Restricting User Interface to Display Custom Test Scripts Only

Most Pc-Check for Windows users will find it easier using customized test scripts to standardize tests processes carried out by technicians.

Once these custom test scripts and configuration files are created and stored, it is possible to deploy them in many alternative ways by combining test scripts, configuration files and command files as described in section named "Understanding Pc-Check Windows Testing Architecture". In this way, an organization can deploy an unlimited amount of combinations to its service organization in addition to the default test scripts supplied by Eurosoft (UK) Ltd and the ad-hoc test execution capability offered by the user interface.

It is also possible to restrict Pc-Check for Windows user interface so that it only displays predefined test scripts and no other testing options are presented to the user. In this way an organization can set up its own set of test sequences and deploy them into all of its branches and technicians. As no other tests are available on the user interface, this forces usage of approved test scripts only. A sample user interface as experienced by a field technician is shown below:

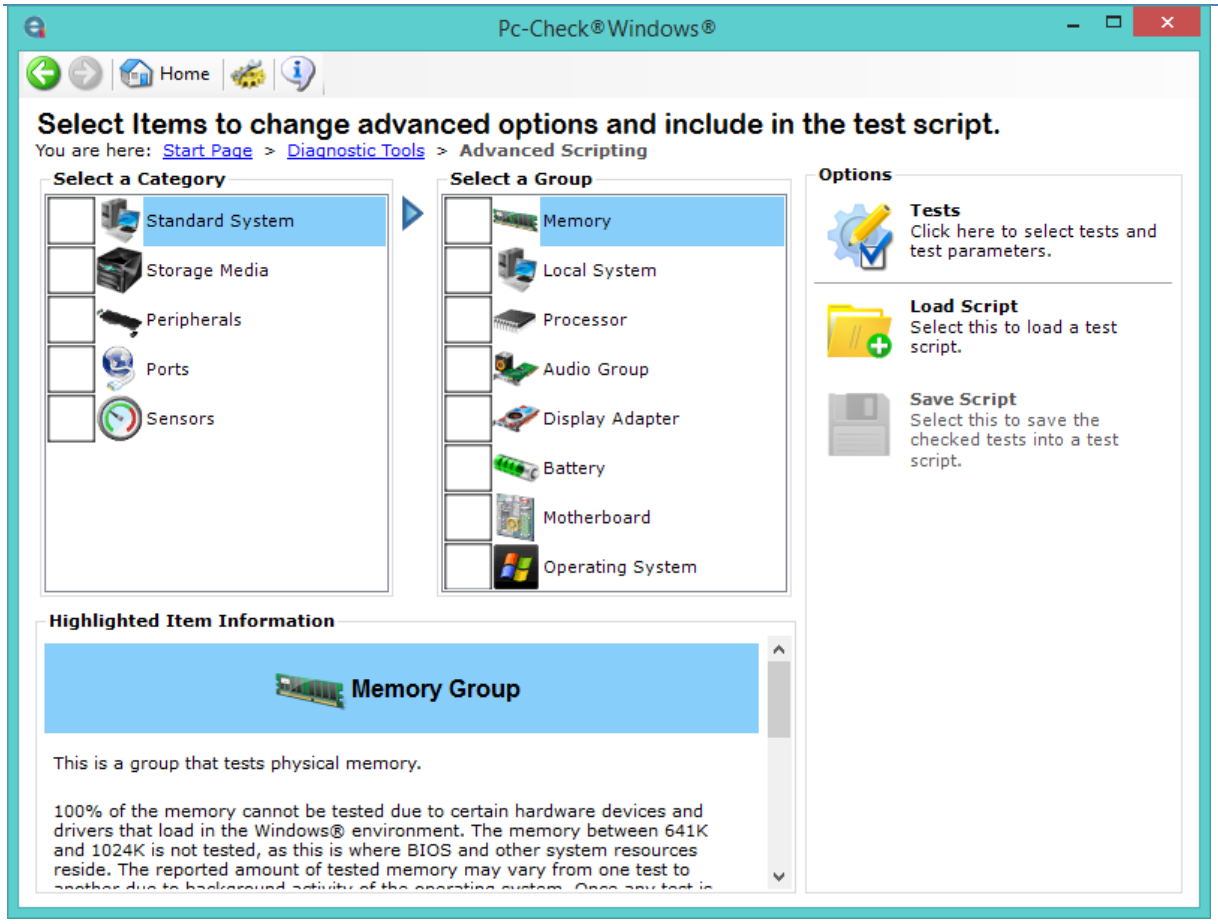


This section will cover the following operations through use of a sample settings:

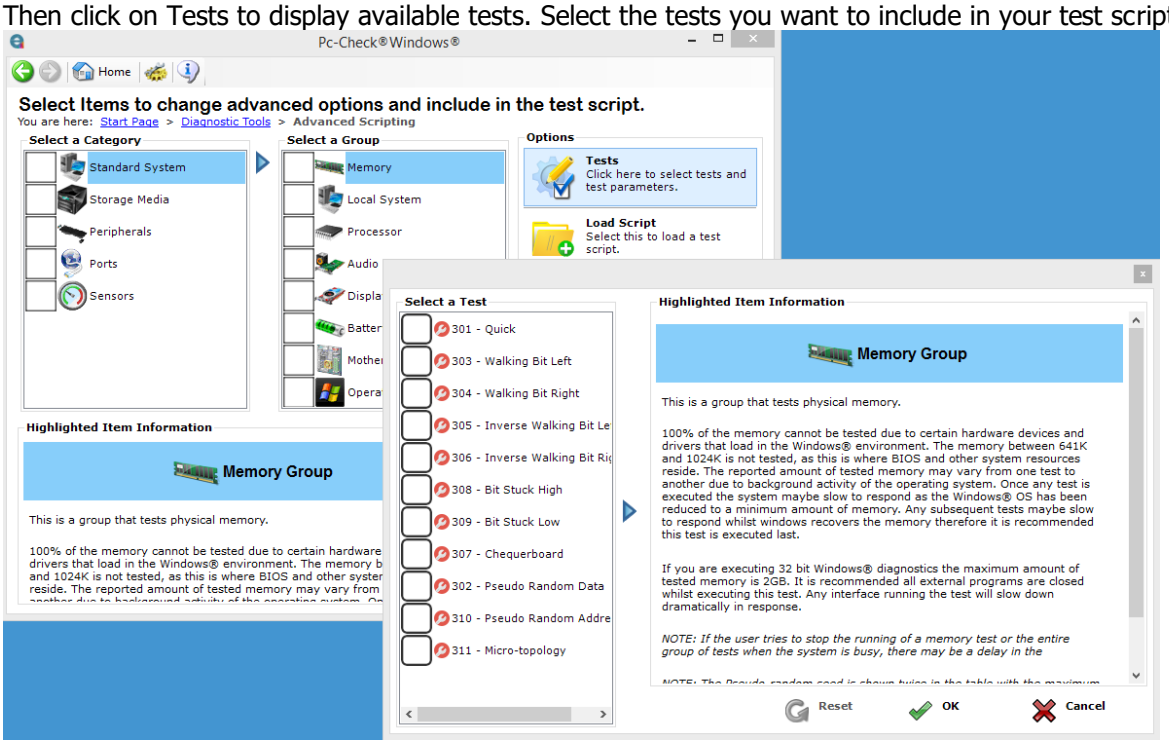
- Creating test scripts to be deployed to organization
- Setting the Interface Visibility Level parameter in related configuration file
- Setting usage of correct configuration file in related command file
- Seeing the resulting user interface at work

Creating test scripts to be deployed to organization

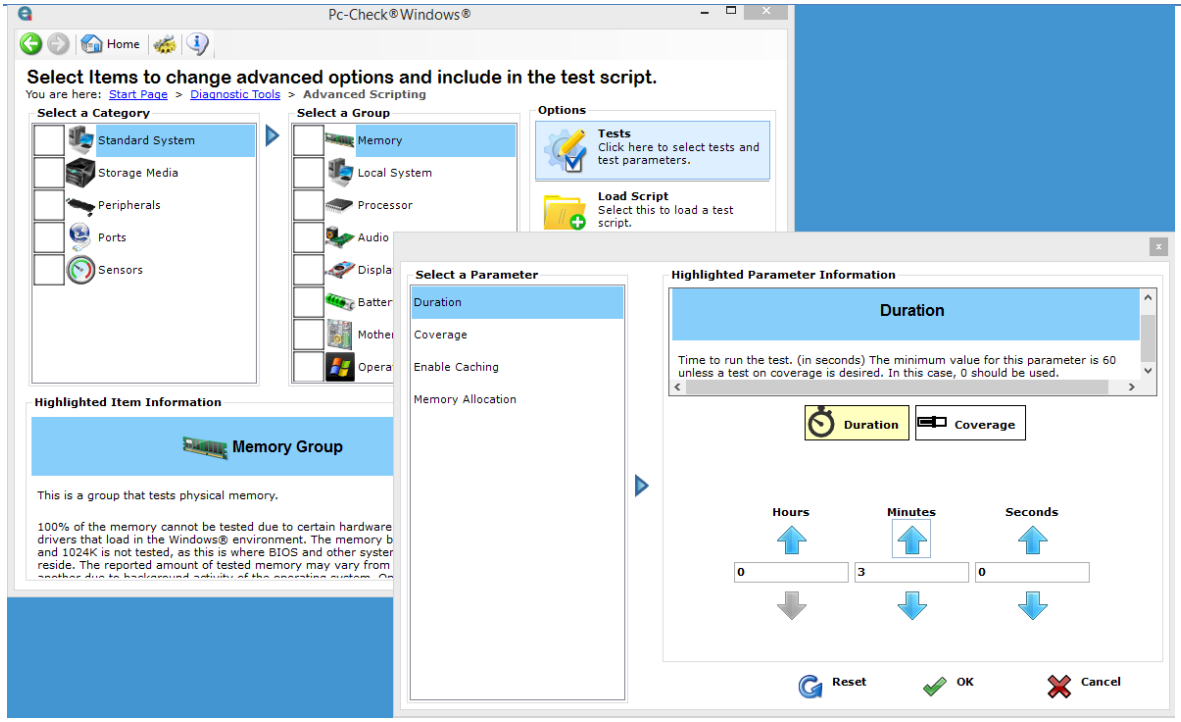
Browse to Start Page-> Diagnostic Tools -> Advanced Scripting screen. Select the Category and Group of the test you want to include in your script.



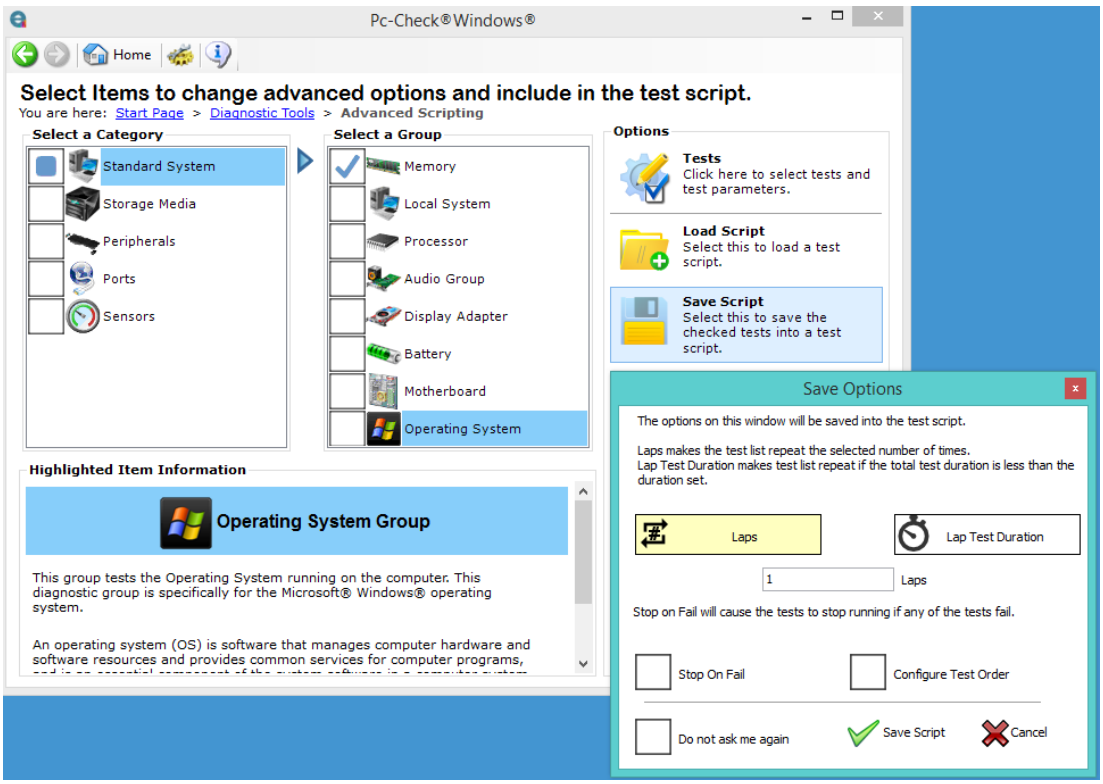
Then click on Tests to display available tests. Select the tests you want to include in your test script as shown below.



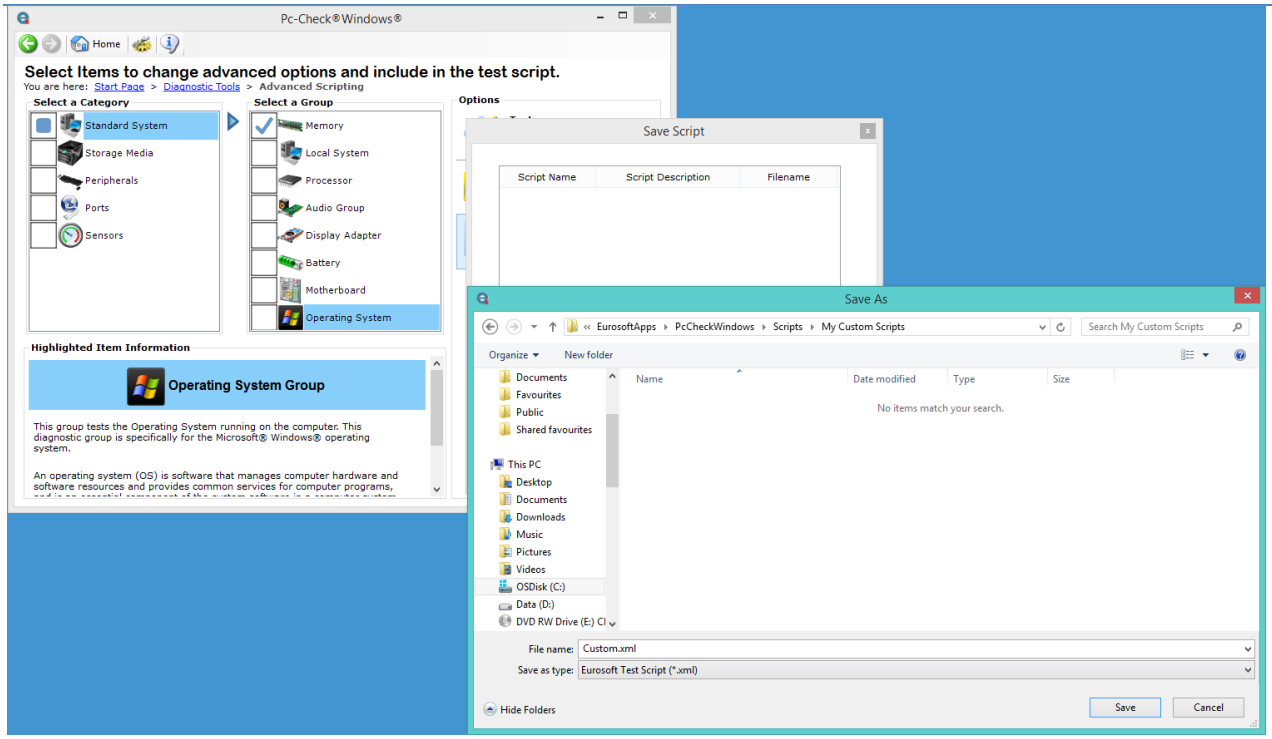
If you want to change the default test parameters, click on the red circle displayed before the test name. Then click on required Parameter name to select from the left panel and make necessary changes on right panel you require and click OK.



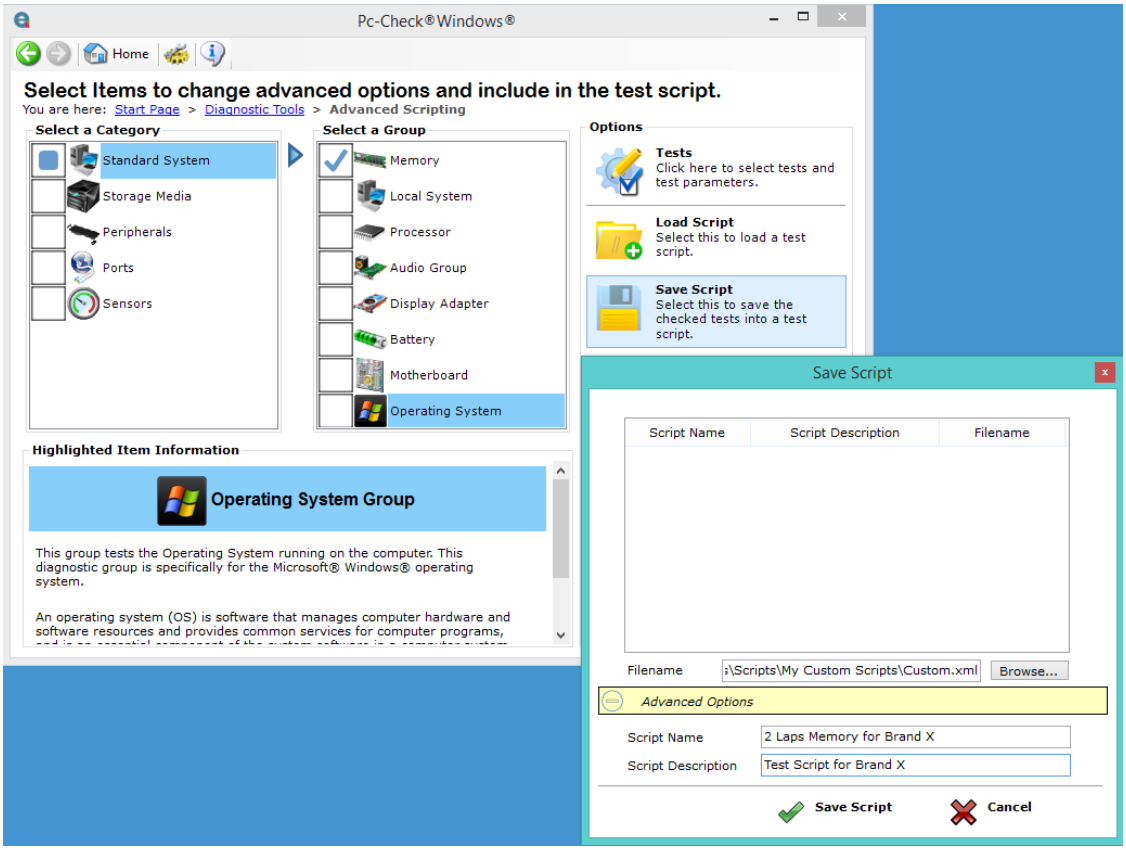
Repeat as many times as necessary to add test categories, test groups and tests to your test script. Once done, click on Save Script and select options you want on Save Options dialog box and click on Save Script button.



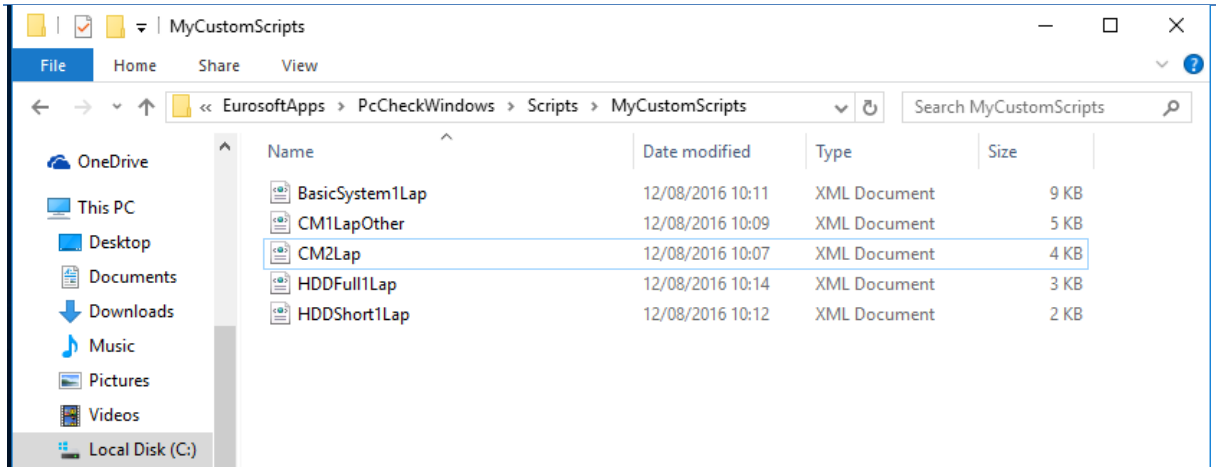
Browse to the folder you want to save your script, enter the name you want to give the script and click on Save button.



Enter Script Name and Script Description information on respective fields. These information will help you to identify test scripts easily while in use later. Click on Save Script button.

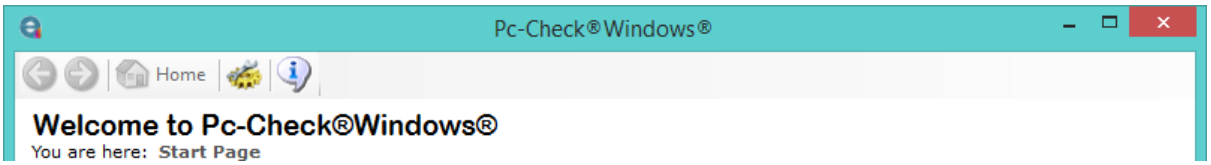


Repeat as many times as necessary until you have all the scripts saved in the folder you want them to be.



Setting the Interface Visibility Level parameter in related configuration file

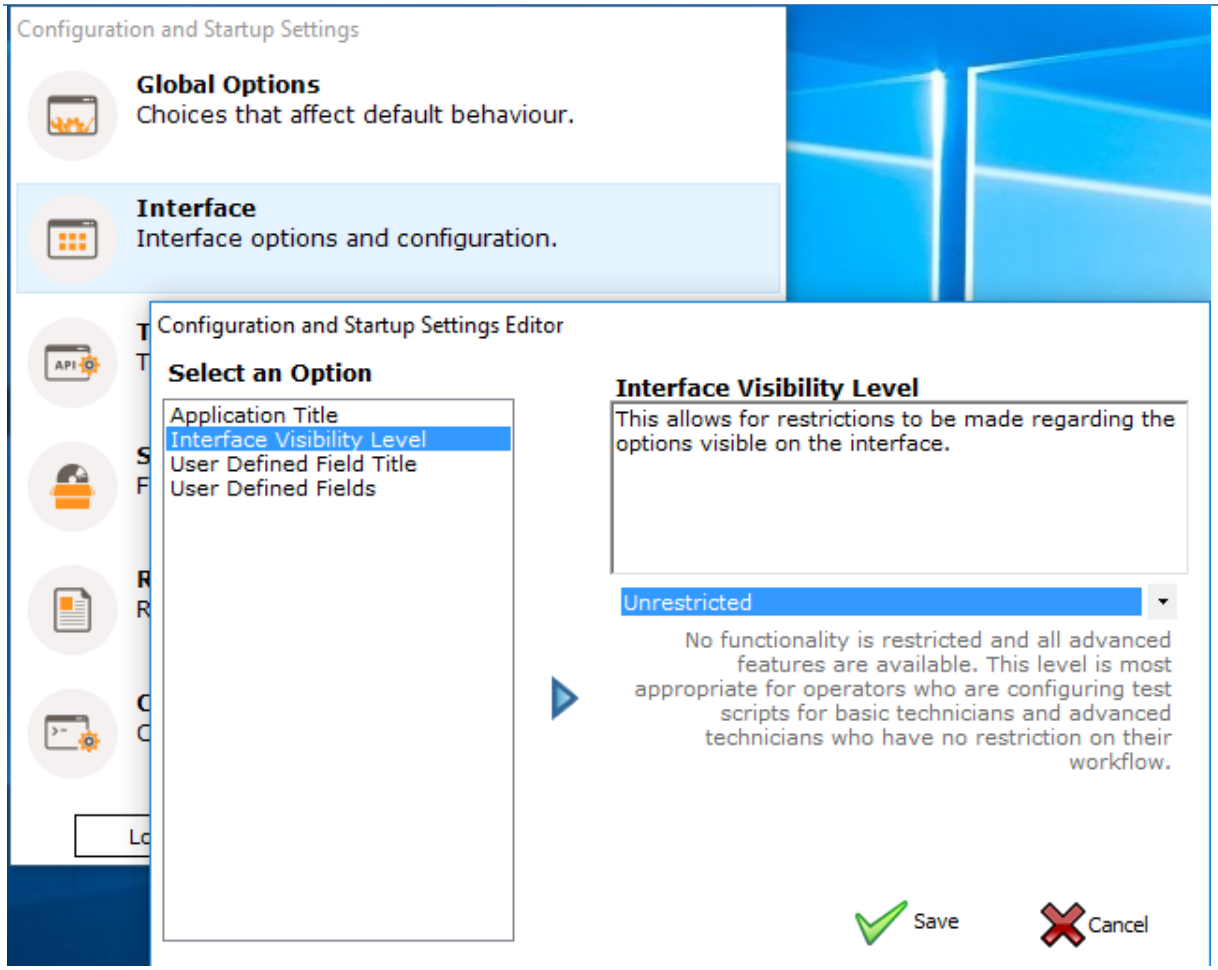
Interface Visibility Level parameter specifies what options are available to user in the interface. To set this parameter, click on Configuration and Startup Settings button that is right next to the Home button.



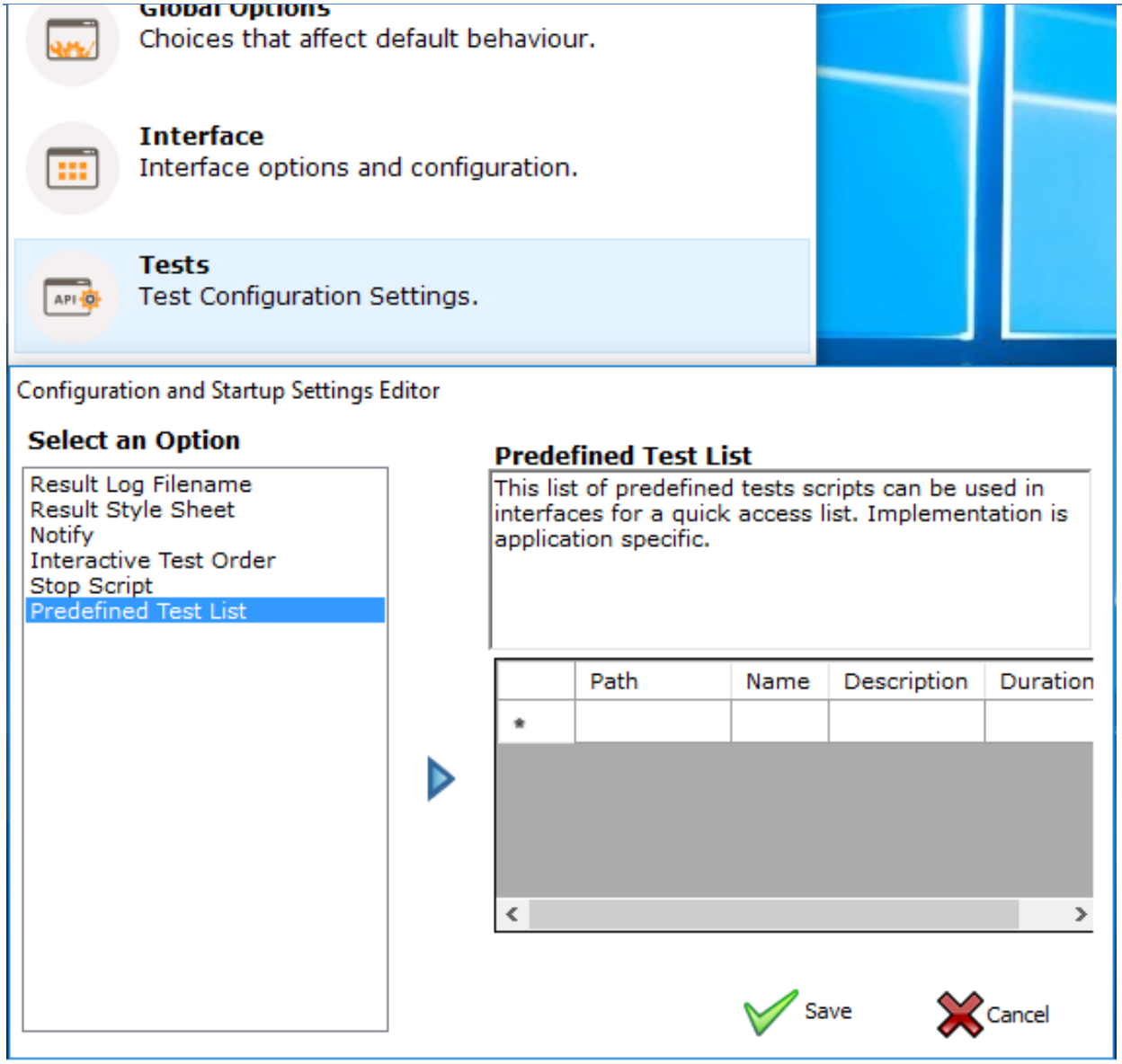
Once the Configuration and Startup Settings window opens, you can select and load the Configuration file you want to edit and set Interface Visibility Level.

Click on Interface, then select Interface Visibility Level from left panel and select the required option from the dropdown box from the right panel.

There are three options you can select: unrestricted, partial and restricted. Select the option your testing process requires and click on Save button.



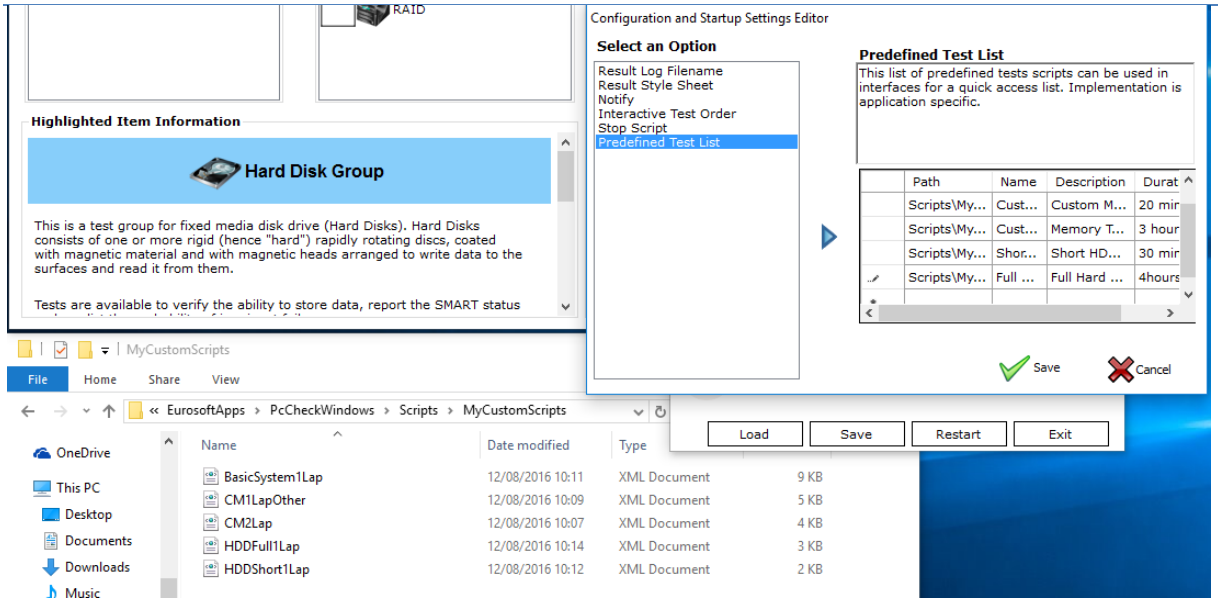
Once you set the Interface Visibility Level, click on Tests and then select Predefined Test List option from the left panel. A tabular entry form will be displayed on the right panel.



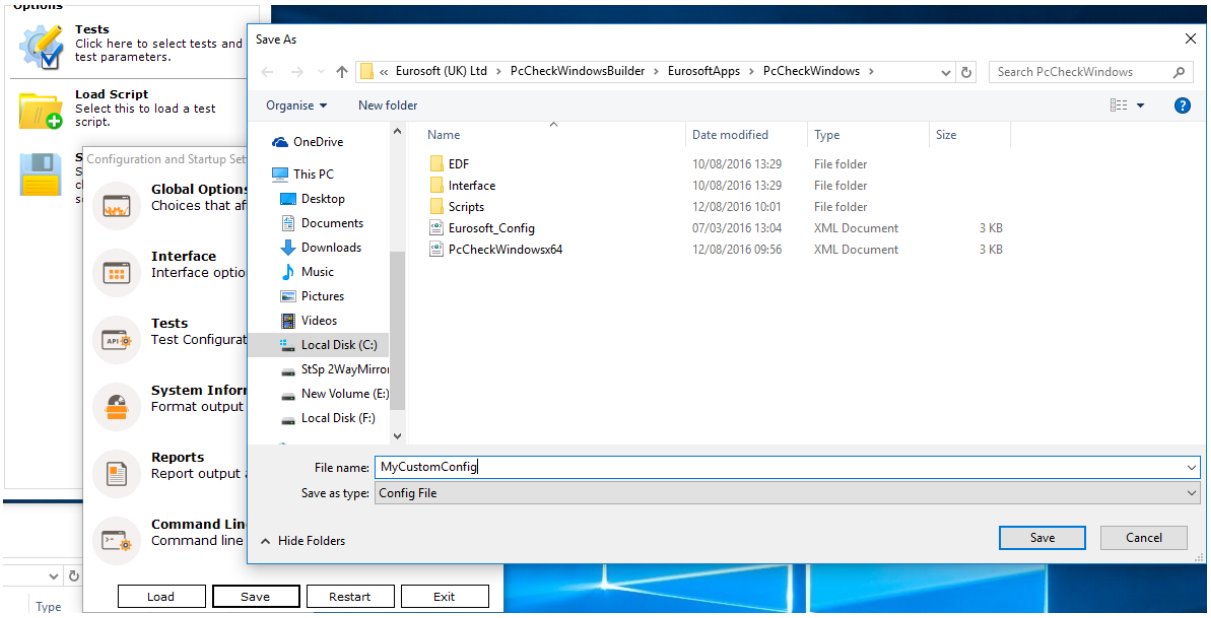
You have to manually type the path of the test script relative to the Pc-Check for Windows root folder, the name you want this script to be displayed to the user, the description you want to be displayed and the approximate duration the test script will take to perform all tests.

While the first two fields are absolutely required for tests to be displayed on the user interface, the last two fields are important to supply the technician with useful information.

Once you fill the information, click on Save button to save your changes to the Predefined Test List.



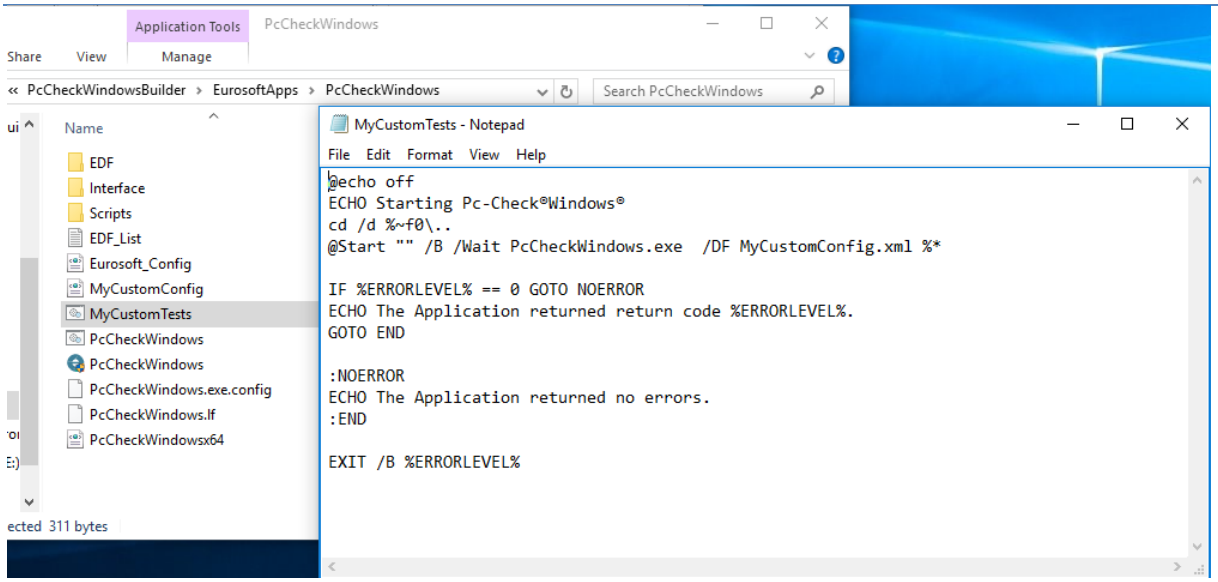
On Configuration and Startup Settings window, click on Save button and enter the name for configuration file you have changed and click on Save.



Setting usage of correct configuration file in related command file

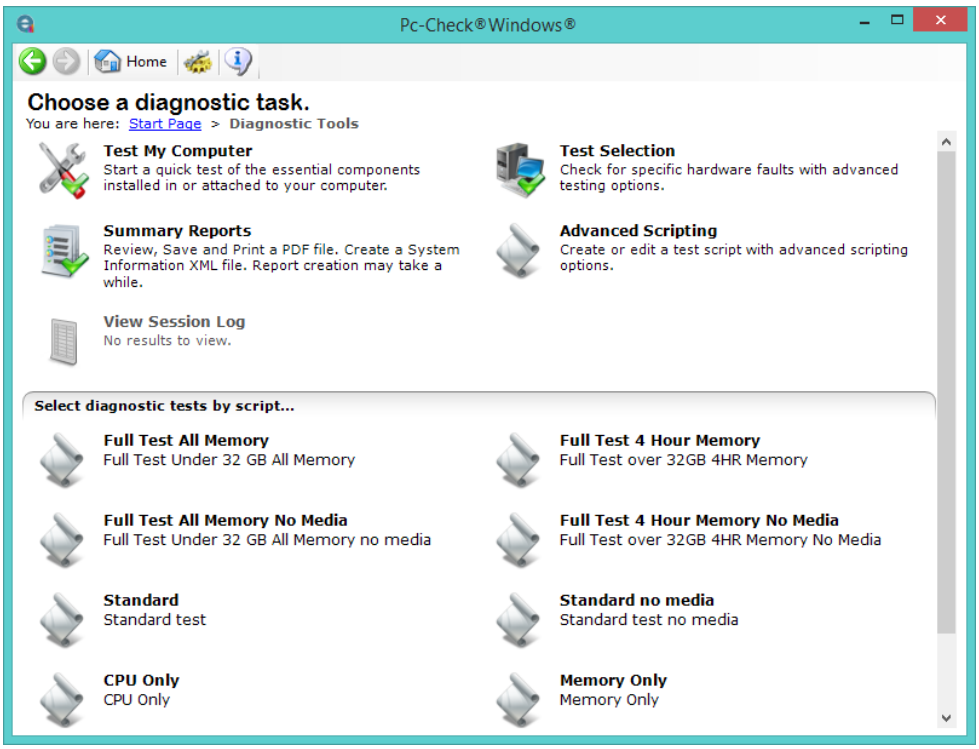
As you know by now, Pc-Check for Windows starts without any default configuration file and test script when it is executed from its exe file. To make it use a specific configuration and test scripts, it must be started from a command file.

Browse to root folder of Pc-Check for Windows and create a copy of PcCheckWindows.cmd file. Rename the copied command file and use your preferred text editor to edit it. Edit the line starting with "@Start …" and change or add the path of your configuration file after "/DF" flag. Save your changes.

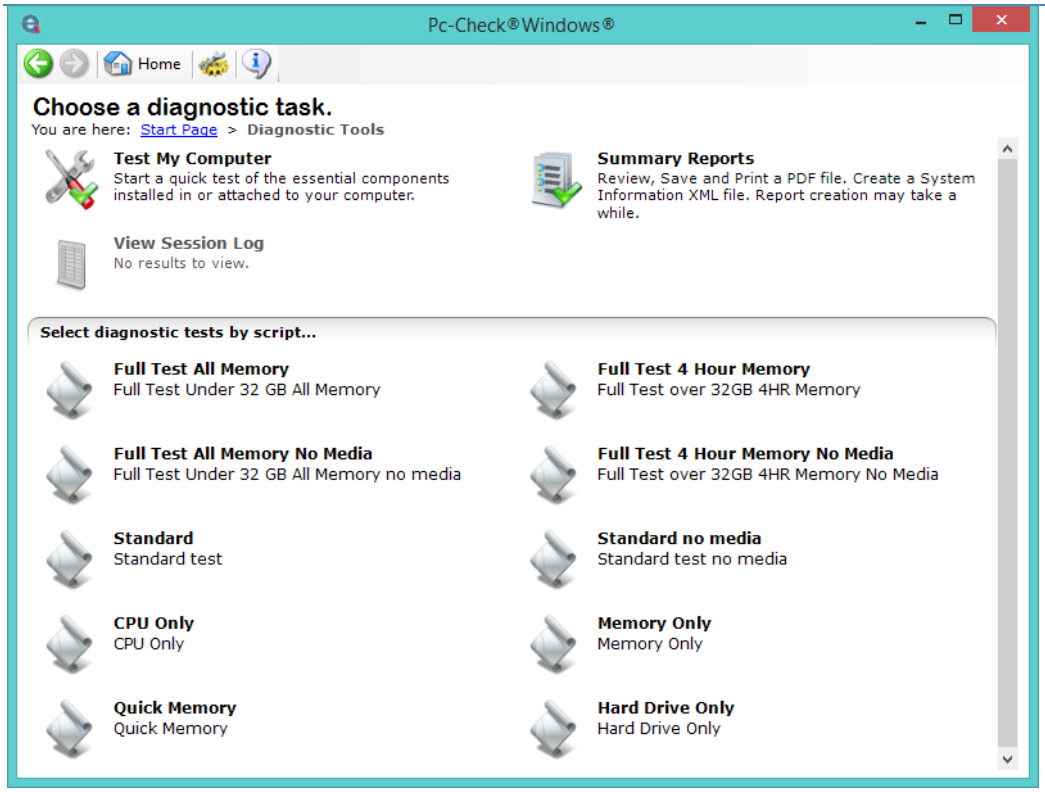


Seeing the resulting user interface at work

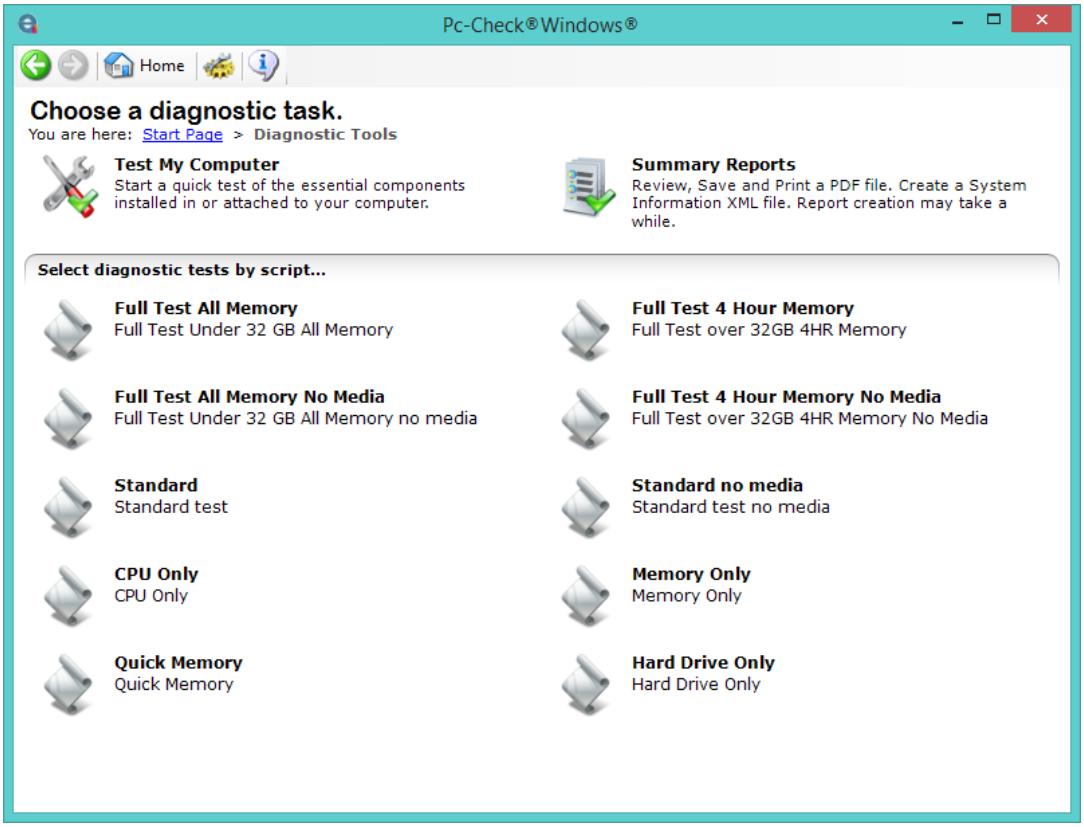
Depending on the Interface Visibility Level you have selected, the Diagnostic Tools window will look different:
If the setting is "Unrestricted", then it will look like this:



If the setting is "Partial", then it will look like this:



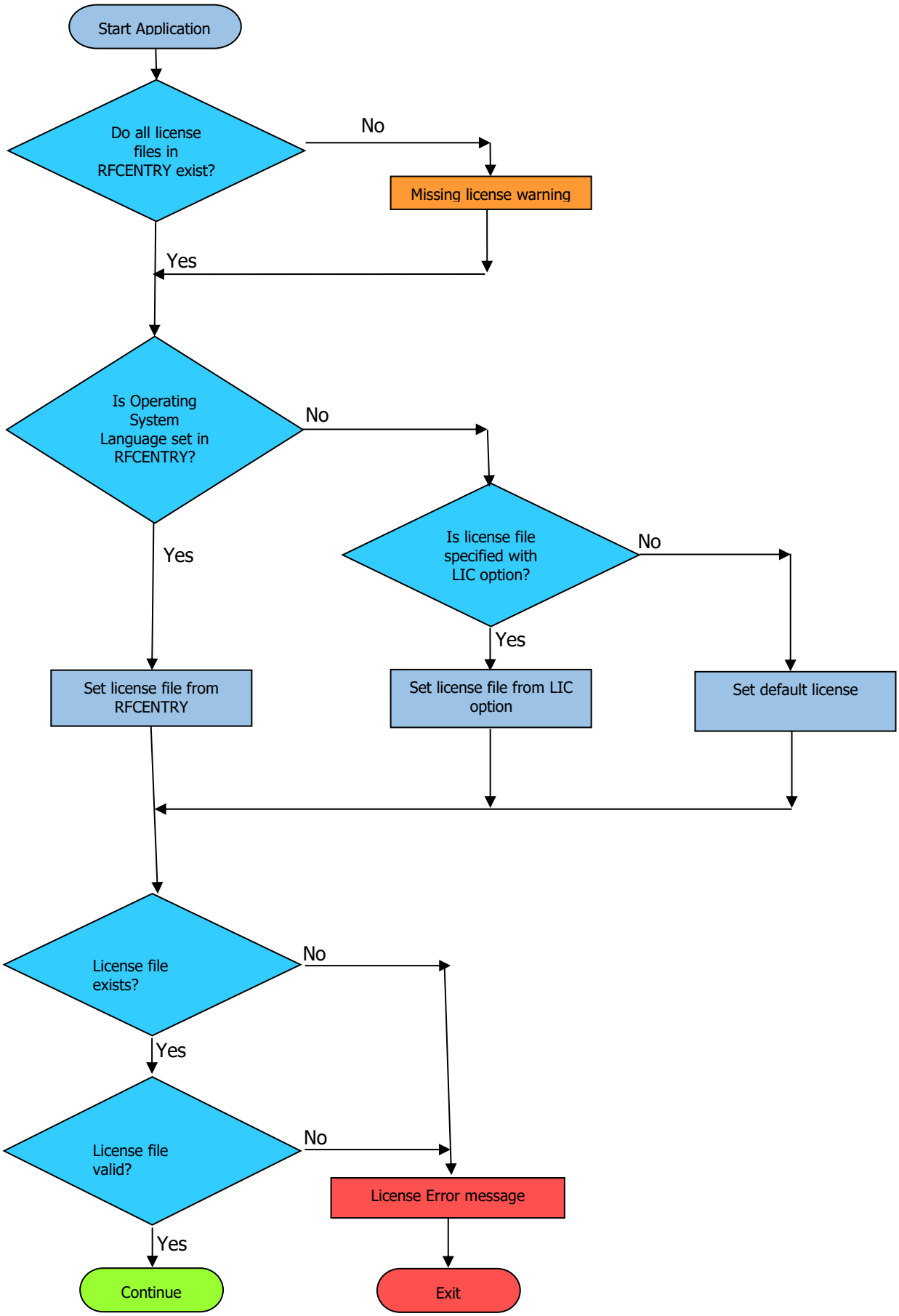
If the setting is "Restricted", then it will look like this:



If you wish Pc-Check Windows to run from a single script use the /R command line option.

For further information about all the available tests and options for Pc-Check Windows please consult the user guide and the test descriptions manual found in the following location:
\Program Files (x86)\Eurosoft (US) Inc\WinPE Image Creator\EurosoftApps\PcCheckWindows\EDF\help

Appendix C: The following flow chart defines the license file set.



Appendix D: **Event Log XML Specification**

The application event log is created on start-up and is named <Title>.xml. This file contains logs of application events such as group initialisation.

EuroEvents Tag

Attribute	Mandatory	Type	Range
XML_VERSION	YES	String	N/A - Fixed at Version 1.0
Version	YES	String	Application Version

Info Tag

Attribute	Mandatory	Type	Range
Description	YES	String	XML String Type

Error Tag

Attribute	Mandatory	Type	Range
Description	YES	String	XML String Type

```
<?xml version="1.0"?>
<EuroEvents XML_VERSION=”1.0” Version=”10.8.0.0”>
        <Info Description="Initialised Parallel Port Tests (group 200)."/>
        <Error Description="Error: Unable to validate &amp;quot;4&amp;quot;. The line 2 was skipped and the default value was used."/>
</EuroEvents>
```

Appendix E: **Result Log XML Specification**

The test related start, result and information events are logged into the Results log. This file is named <Application Title>_Results.xml by default. This can be configured using the FilenameResults tag of the configuration file. See 'Configuration File' for more information.

EuroLog Tag

Attribute	Mandatory	Type	Range
XML_VERSION	YES	String	N/A - Fixed at Version 2.0
Version	YES	String	Application Version

LogEvent Tag

This is a subtag of Eurolog.

Attribute	Mandatory	Type	Range
DateTime	YES	[-]CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]	ISO 8601
Group	YES	Integer	1-9999
GroupName	NO	String	N/A
Test	YES	Integer	1-9999
TestName	NO	String	N/A
Device	YES	Integer	1-9999
DeviceName	NO	String	N/A
Message	YES	See msg table below (Integer)	2-18
LapCount	YES	Integer	1-999
Description	NO	String	See note below
pferror	YES	String	N/A
pfile	YES	String	N/A

Note: *The description attribute is not mandatory but if it is populated additional information may be provided. In these cases the string takes the format for an error <x>-<y>-<z> where y is the specific diagnostic group error code and z contains extra information. Detailed information on the format of these codes can be found in 'Advanced Options – Error Codes and Extra Information' section.*

Message Table

Integer Value	Meaning
2	Test Start
3	Test Passed
4	Test Skipped
5	Test Failed

6	Test Aborted
7	Test Not Available

LogInfo Tag

This is a subtag of Eurolog.

Attribute	Mandatory	Type	Range
DateTime	YES	[-]CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]	ISO 8601
Group	YES	Integer	1-9999
Test	YES	Integer	1-9999
Device	YES	Integer	1-9999
Message	YES	Integer	See msg table below (Integer)
extra	YES	Integer	Content depends on Message
Description	NO	String	N/A

Message Table

Integer Value	Meaning
0	Media Read Failed – extra contains location in hexadecimal
1	Drive Failure – Extra contains the hardware IDs for the failed driver
2	Windows® Event Failed – Extra contains the failed windows® event ID

```
<?xml version="1.0"?>
<EuroLog XML_VERSION=”2.0” Version="10.6.0.0">
	<LogEvent datetime="2013-10-29T11:45:25+01:00" group="2601" test="304"
device="1" message="2" lapcount="1" pferror="0000000000000000" pffile=""
groupname="Hard Disk" testname="S.M.A.R.T. Status" devicename="Hard Disk 0"
description="2601 - 1 - 304"/>
	<LogEvent datetime="2013-10-29T11:45:25+01:00" group="2601" test="304"
device="1" message="3" lapcount="1" pferror="0000000000000000" pffile=""
groupname="Hard Disk" testname="S.M.A.R.T. Status" devicename="Hard Disk 0"
description="PASSED - 0000000000000000"/>
</EuroLog>
```

Appendix F: Script XML Specification

Introduction

Device Attributes and all other mandatory sub attributes are only such if the Device Element is used.

Tag and Attribute values

Script Tag

Attribute	Mandatory	Type	Range
XML_VERSION	YES	String	N/A – Fixed at 3.0
RunType	YES	String	"Duration|Lapcount"
RunCount	YES	Integer	1-9999
StopOnFail	NO	Boolean	0-1

ScriptInfo Tag

The ScriptInfo Tag is an optional tag that can be used in the root of the XML script to specify the Name, Duration and Description of the script. This can be used by some interfaces to provide more information when selecting tests.

Attribute	Mandatory	Type	Range
Name	YES	String	N/A
Desc	NO	String	N/A
Time	NO	String	N/A

Group Tag

Attribute	Mandatory	Type	Range
Group	YES	Integer	1-9999
GroupName	NO	String	N/A

Device Tag

This is a subtag of Group. A value of 0 indicates all devices within the group.

Attribute	Mandatory	Type	Range
Device	YES	Integer	0-9999

Test Tag

This is a sub tag of Device.

Attribute	Mandatory	Type	Range
Test	YES	Integer	1-9999
TestName	NO	String	N/A

TestParams Tag

This is a mandatory tag that must be present in all scripts. It has no values.

TestParam Tag

This is an optional sub tag of test. The tests run in order sequence and for these to take effect they must be setup beforehand. See the Test Descriptions document for more details.

Attribute	Mandatory	Type	Range
ID	YES	Integer	1-9999
Name	NO	String	N/A
Value	YES	Multi	Type depends on test param ID

Example Test Script

Typical example of text script:

```
<?xml version="1.0" encoding="UTF-8"?>
<Script XML_VERSION=”3.0” RunType="LapCount" RunCount="1" StopOnFail="0">
<ScriptInfo Name="Quick Test" Desc="Eurosoft Quick test script" Time="5 mins"/>
<Group Group="260X" GroupName="Hard Disk">
        <Device Device="0">
                <Test Test="303" TestName="Linear Read">
                <TestParams>
                        <TestParam ID="1" Name="Duration" Value="60"></TestParam>
                        <TestParam ID="2" Name="Coverage" Value="100"></TestParam>
                        <TestParam ID="3" Name="MaximumErrors"
Value="1"></TestParam>
                </TestParams>
                </Test>
        </Device>
</Group>
</Script>
```

Appendix G: **System Information Log XML Specification**

When the application is run with the SXML command line option or SXML 1 configuration option, the system information is collected and output into the System Information.

System Profile Tag

Attribute	Mandatory	Type	Range
XML_VERSION	YES	String	N/A - Fixed at Version 1.1

Eurosoft Tag

This is a sub Tag of the SystemProfile tag and defines a diagnostic group. The Group attribute specifies the Group's ID.

Component

This is a sub tag of the Eurosoft tag and defines a device within a group. The ID attribute specifies the Device ID. The DeviceCode attribute specifies the device type code.

This tag can be defined within Component tags. In this case the component represents a sub device.

Attribute Tag

This is a sub tag of the Component ta. The Attribute ID is the unique identifier for the attribute within the group. The name attribute is the device attribute name. The value attribute is the device attribute value.

```
<?xml version="1.0" encoding="UTF-16" ?>
<SystemProfile>
	<Eurosoft Group="1">
		<Component ID="0" DeviceCode="0">
			<Attribute ID="0" Name="Tester" Value="Engineer32"></Attribute>
			<Attribute ID="1" Name="Machine" Value="Machine32"></Attribute>
		</Component>
	</Eurosoft>
	<Eurosoft Group="201">
		<Component ID="1" DeviceCode="17">
			<Attribute ID="5242881" Name="Device" Value="0"></Attribute>
			<Attribute ID="5242883" Name="PNPDeviceID"
Value="NULL"></Attribute>
			<Attribute ID="5242884" Name="DeviceID"
Value="NULL"></Attribute>
			<Attribute ID="1" Name="BaseIOAddress" Value="0"></Attribute>
		</Component>
	</Eurosoft>
```

```
</SystemProfile>
```

XML Style Sheet

Included in the interface directory are two style sheets that gives an example of how the results xml file and Sysinfo.xml can be interpreted and formatted. The /RXSL and /SXSL command line options can be used to select the style sheet that should be used for these files.

These style sheets are an example of the potential functionality only and therefore are not directly supported by Eurosoft.

The style sheets are commented in relevant places to assist the user understanding the XSL code. These comments are only supplied in English.

Note: *The /LOG and /SXML filenames can be dynamically configured using the configuration file. Information on dynamic filenames can be found in the 'Dynamic Fields' section.*

Eurosoft (UK) Ltd
Head Office
3 St Stephens Road
Bournemouth, UK
BH2 6JL
Tel +44 (0)1202 297315
Fax +44 (0)1202 558280
info@eurosoft-uk.com
www.eurosoft-uk.com

Eurosoft (US) Inc.
North American Sales Office
706 Jackson Street,
Sioux City, IA 51105, USA
Tel US Toll Free +1 (888) 980-9595
Tel +1 (712) 255 7483
Fax +1 (866) 615-9384
info@eurosoft-us.com
www.eurosoft-us.com

Note: Windows PE contains a security feature that will cause end user's systems to reboot without prior notification to the end user after 72 hours of continuous use.