- Link Check for Windows 3.x, Windows 95 and Windows NT
- Version 5.70 - October 1997
- Copyright (C) Karri Software Ltd. 1994-1997.

Contents for Link, Memory and Function Check Help

For latest information, please visit our Web page
http://ourworld.compuserve.com/homepages/KarriSL

Please select one of the following:

What's New

Major changes from Version 5.30 to Version 5.70

- Now fully functional
- All version information now shown
- Clipboard copy added for all info in tab-delimited form for easy export
- "Ignore Modules in Memory" Option added
- "Show Self-Registering Executables" Option added for identification of those files (32-bit only)
- New interface
- .OCXs are now shown correctly in the Windows NT version
- New reports
- New "Memory Details" display
- Old executable (MZ) header display added
- New executable (NE) header display added
- Portable executable (PE) header display added (32-bit only)
- Improved toolbar
- Function Check displays can now be sorted
- Ordinals in Function Check can now be displayed in decimal/hexadecimal
- CTL3D32.DLL fix for Windows NT version
- Can now self-register and unregister DLLs from the Registry (32-bit only)
- Link Check can now create a response file for zipping purposes
- Evaluation period extended to 30 days
- Help files merged into just the one (WLCHECK.HLP)
- Other minor bugs fixed

Ordering Information

Prices for **Link Check** are as follows:

Link Check for Windows 3.x	**$29**
Link Check for Windows 95	**$35**
Link Check for Windows NT	**$35**
Link Check Suite (all three above)	**$50**

Site licenses are also available at $25/$30/$30/$45 respectively for more than 10 purchases.

The package also contains **Memory Check** and **Function Check** for no extra charge.

You may order **Link Check** via:

CompuServe
NorthStar Solutions

CompuServe automatically adds the registration fee to your monthly bill.

NorthStar Solutions accept credit cards, checks and money orders via the WWW, Telephone, Fax, Email or Post.

You may also order via the Web through our homepage **http://ourworld.compuserve.com/homepages/KarriSL**.

The moment either of the above notify us of a successful registration, we will email you the latest retail version of the registered product.

Ordering via CompuServe

Ordering through CompuServe's Software Registration facility SWREG is quick and easy.

Log onto CompuServe and **GO SWREG**. Follow the instructions on screen and use one of the following registration IDs when prompted:

Link Check for Windows 3.x	**ID 7160**
Link Check for Windows 95	**ID 10627**
Link Check for Windows NT	**ID 10094**
Link Check Suite (all three above)	**ID 10095**

CompuServe will automatically email us with a registration notification and once upon receipt we will email the license file to you.

Ordering via NorthStar Solutions

Ordering through NorthStar Solutions is quick and easy. Please quote one of the following product IDs when ordering:

Link Check for Windows 3.x	**ID 1455**
Link Check for Windows 95	**ID 1456**
Link Check for Windows NT	**ID 1457**
Link Check Suite (all three above)	**ID 1458**

INTERNET ORDERS
Visit NorthStar Solutions at
http://nstarsolutions.com
and fill out their online order form--fast, easy and secure!

Alternatively, a direct link to the **Link Check** page is **http://nstarsolutions.com/904.htm**.

PHONED ORDERS
Calls are taken 10 am - 8 pm, EST, Monday thru Saturday.
1-800-699-6395 (From the U.S. only.)
1-803-699-6395

FAXED ORDERS
Available 24 hours. International & business orders are encouraged.
1-803-699-5465

E-MAILED ORDERS
CompuServe: starmail
America Online: starmail
Internet: starmail@compuserve.com

MAILED ORDERS
You may register with a check or money order.
Make them payable to **"NorthStar Solutions"** and send them to:
PO Box 25262, Columbia, SC 29224, USA

Please provide (or be prepared to provide) the following information:
* The program you are registering and its ID (found at the top of this page)
* Your mailing address
* Your Visa, MasterCard, or Discover # and its expiration date (if using credit card)
* Your E-Mail address (so NorthStar Solutions can send you an E-Mail confirming your order and so we can contact you easily with the license file and any important follow-up information, upgrade announcements, etc.).

NorthStar Solutions will automatically email us with a registration notification and once upon receipt we will email the license file to you.

Introduction

Link Check is a suite of three (3) diagnostic programs which allows the user to examine different areas of the system. These are useful not only on solving system problems but also for learning about the system.

All three programs show version and file information on any of the modules (files) for identification purposes as well as interacting with each other.

Armed with these three, you should be able to solve most of your software problems:

Link Check

Link Check enables the user to view the links between an executable file and the modules it requires to run on the system. This tool is useful for analyzing those "Cannot find (filename) or one of its components" system error messages.

You may also use **Link Check** to view which versions of modules (DLLs, VBXs, OCXs) the program will actually use in the system as there are duplicate DLLs frequently lying about the network or the machine.

Link Check mimics the system when a program is finally executed by the operating system (Windows). This will help in determining beforehand whether a program will run or not, or which particular copy of a .DLL it is going to use if there are multiple copies lying around the system.

Any file(s) required by the program that are missing from the system will be highlighted with a red exclamation mark. It could be that the file(s) are actually present on the system but will not get picked up by the system during run time. This could be because the system environment variable PATH has been changed.

Link Check is also useful for analyzing software installation problems as well as indicating which support modules should be backed-up with a particular program.

Memory Check

Memory Check allows the user to view, load and unload modules currently in memory. You may also view the current status of memory, including the amount of free memory left.

Important version information about each module can be displayed and you may create and print a summary report of the state of the system.

Memory Check really comes into play when some program has ended abnormally and has left modules lying around in memory. The system might behave irratically or more to the point, those modules take up valuable memory. In these situations you can use **Memory Check** to unload those modules and save yourself a reboot of Windows.

Warning: Sometimes a clean unload is not possible and you may find yourself unceremoniously dumped back on to the C: prompt, so please do use care when unloading.

You may also verify exactly which version of a module is currently loaded in memory. Sometimes some programs will not run because a wrong version of a particular module (.DLL) has already been loaded into memory by some other application. Windows does not

load a new copy of a file with the same file name but will simply increase the "usage count" of the module already in memory. These "usage counts" are displayed by **Memory Check** as a number in parenthesis () after the file name (not in the Windows NT version though).

WINSOCK.DLL is a very good example of a module which shares very many different versions. Most communications programs require their own particular version of WINSOCK.DLL to be loaded, but if another application has put their version in first, it is highly likely that the second one will fail or behave in manner not designed by the programmers. Use **Memory Check** to solve these type of problems.

Function Check

Function Check allows the user to view actual function calls inside modules. This program is useful for analyzing those "Call to Undefined Dynalink" system error messages.

When a program calls a module (.DLL) to execute a function (piece of code) inside the .DLL and that particular function does not exist in the .DLL, a "Call to Undefined Dynalink" error is raised by the system. These can occur when a newer version of a .DLL is accidentally replaced by an older version of the same .DLL. A program is compiled to use the new functionality of the .DLL but can come unstuck when that .DLL is replaced by an older version which does not contain the new functionality.

Some installation programs are prone to do this "accidental" replacement.

Function Check cross-checks all function calls in an executable with the ones in all of the .DLLs (modules) that it uses, to verify that all of the functions all present.

Should your program throw up one of those "Call to Undefined Dynalink" messages, **Function Check** will show you which module is at fault.

System Requirements, Installation and Uninstall

It is recommended that all of the files contained in this package are unzipped to a single directory such as C:\LINKCHK. This will avoid any future versioning problems between different releases of the program(s).

The Windows 95 version does not run under Windows NT due to Kernel differences between the two operating systems. Please use the NT version instead.

Link Check for Windows 3.x requires Windows 3.1* or later or Windows for Workgroups 3.11*.
Link Check for Windows 95 requires Windows 95* or later.
Link Check for Windows NT requires Windows NT 3.51* or later.

* See Trademarks

The 16-bit Windows 3.x package consists of the following files:

1) WLCHECK.EXE	**Link Check** itself
2) WLCHECK.HLP	Link Check Help file for all three programs
3) WMCHECK.EXE	**Memory Check** component of Link Check
4) WFCHECK.EXE	**Function Check** component of Link Check
5) WLCCOMM.DLL	Common Routines for all of the Link Check components
6) LCHOOKS.DLL	More Common Routines for the Link Check components
7) FILE_ID.DIZ	File for Bulletin Board operators
8) README.TXT	Read Me file

To **uninstall Link Check** delete the above files and the following:

1) WLCHECK.INI	Created by Link Check in the Windows directory
2) WMCHECK.INI	Created by Memory Check in the Windows directory
3) WFCHECK.INI	Created by Function Check in the Windows directory

The 32-bit Windows 95 package consists of the following files:

1) WLCHK95.EXE	**Link Check** itself
2) WLCHK95.HLP	Link Check Help file for all three programs
3) WMCHK95.EXE	**Memory Check** component of Link Check
4) WFCHK95.EXE	**Function Check** component of Link Check
5) WLCCOM95.DLL	Common Routines for all of the Link Check components
6) LCHKS95.DLL	More Common Routines for the Link Check components
7) FILE_ID.DIZ	File for Bulletin Board operators
8) README.TXT	Read Me file

To **uninstall Link Check** delete the above files.

The 32-bit Windows NT package consists of the following files:

1) WLCHKNT.EXE	**Link Check** itself
2) WLCHKNT.HLP	Link Check Help file for all three programs
3) WMCHKNT.EXE	**Memory Check** component of Link Check
4) WFCHKNT.EXE	**Function Check** component of Link Check
5) WLCCOMNT.DLL	Common Routines for all of the Link Check components
6) LCHKSNT.DLL	More Common Routines for the Link Check components
7) PSAPI.DLL	Functions for Memory Check
8) IMAGEHLP.DLL	Functions for PSAPI.DLL
9) FILE_ID.DIZ	File for Bulletin Board operators
10) README.TXT	Read Me file

To **uninstall Link Check** delete the above files.

Command Line Interface and Drag & Drop

Link Check accepts a file name as a command line parameter.

Win 3.x Syntax: **WLCHECK Filename**
Win 95 Syntax: **WLCHK95 Filename**
Win NT Syntax: **WLCHKNT Filename**

Memory Check has no command line parameters.

Function Check accepts a file name as a command line parameter and a switch for import/export view.

Win 3.x Syntax: **WFCHECK Filename [/E | /I]**
Win 95 Syntax: **WFCHK95 Filename [/E | /I]**
Win NT Syntax: **WFCHKNT Filename [/E | /I]**

where **/E** means show **Exported** functions upon entry and **/I** means show **Imported** functions upon entry.

Link Check and **Function Check** also support **Drag and Drop**. You may drag a file to **Link or Function Check** and drop it in.

Guide to the programs

Link Check

Use the **Open Executable** menu option under **File** to open up a module for examination.

The main window shows the links between the opened executable and the modules that it requires to be present on the system.

If you select any one of the shown modules and **double-click** it with the mouse, **Link Check** reveals further links for the selected module.

To identify any particular module, select it with the mouse and choose the **Display Version Information** menu option under **Options**. A version information window will appear at the right-hand side of the main window.

Each module is prefixed with a symbol identifying the module type. To find out what these symbols mean, please select the **Display Key to Symbols** menu option under **Options**. The key window will appear at the right-hand side of the main window.

You may also view any of the executable headers for the selected file from the same **Options** menu.

To hide any of the information windows, please re-select the same menu items as for invoking them. Alternatively, **right-click** with the mouse on top of these windows and select the **Hide** menu option from the popup menu that follows.

To copy the version information to the clipboard, **right-click** on the window to access the **Copy** menu option or use the short cut off the toolbar.

As a short cut, you may also **right-click** on the main window to access the options in the **Options** menu.

If you prefer to use the keyboard, please see Keyboard Support for the key assignments.

Memory Check

The **Memory Check** main window shows all modules currently loaded in memory.

If you select any one of the shown modules and **double-click** it with the mouse, **Memory Check** reveals version information for the selected module.

This is identical to selecting it with the mouse and choosing the **Display Version Information** menu option under **Options**. In both cases a version information window will appear at the right-hand side of the main window.

Each module is prefixed with a symbol identifying the module type. To find out what these symbols mean, please select the **Display Key to Symbols** menu option under **Options**. The key window will appear at the bottom of the main window.

A number (usage count) is shown in brackets **()** which indicates the number of references made to that module by others in the system.

You may also view any of the executable headers for the selected file from the same

Options menu.

To view only certain file types, you may select the **Set Filters** menu option under **Options**. For further information please see Set Filters dialog box.

To hide any of the information windows, please re-select the same menu items as for invoking them. Alternatively, **right-click** with the mouse on top of these windows and select the **Hide** menu option from the popup menu that follows.

As a short cut, you may also **right-click** on the main window to access the options in the **Options** menu.

If you prefer to use the keyboard, please see Keyboard Support for the key assignments.

Function Check

Use the **Open Executable** menu option under **File** to open up a module for examination.

The **Function Check** main window shows all imported or exported function names with their ordinals (indices) in an executable file. Functions from other modules (such as .DLLs) can be imported by name or by ordinal.

If you select any one of the shown modules (while viewing **imported** functions) and **double-click** it with the mouse, **Function Check** reveals version information for the selected module otherwise for the loaded/open module.

This is identical to selecting it with the mouse and choosing the **Display Version Information** menu option under **Options**. In both cases a version information window will appear at the right-hand side of the main window.

You may also view any of the executable headers for the selected file from the same **Options** menu.

To hide any of the information windows, please re-select the same menu items as for invoking them. Alternatively, **right-click** with the mouse on top of these windows and select the **Hide** menu option from the popup menu that follows.

As a short cut, you may also **right-click** on the main window to access the options in the **Options** menu.

When viewing **exported** function names, the module name is missing since all the functions belong to the currently open module.

Use the sort toolbar buttons to sort the ordinals, function names or module names in either ascending or descending order.

If you prefer to use the keyboard, please see Keyboard Support for the key assignments.

Keyboard Support

The main window for all three programs supports the keyboard as well as the mouse.

Key actions in the **Link Check** main window are as follows:

Tab	Switch between the main and the information windows.
Enter	Expand/Collapse currently selected module.
Spacebar	Select a module.
Arrow keys	Navigate up/down the list of modules.
Control+E	Open executable for examination.
Control+O	Open an ODBC driver for examination.
Control+V	Display Version Information window.
Control+K	Display Key window.
Control+C	Copy information to the clipboard.
Control+P	Print information/report.
Control+R	Run executable.
Control+X	Expand all levels.
Control+L	Expand one more level.
F1	View Help.
F2	Check exported function calls in the selected module.
F3	Check currently loaded modules in RAM (memory).
F4	Check imported function calls in the selected module.
F5	Collapse all open links.
F6	Display old executable (MZ) header.
F7	Display new executable (NE) header.
F8	Display portable executable (PE) header.
Alt+F4	Quit Link Check.

Key actions in the **Memory Check** main window are as follows:

Tab	Switch between the main and the information windows.
Enter	View version information window.
Spacebar	Select a module.
Arrow keys	Navigate up/down the list of modules.
Control+O	Load an executable into memory.
Delete	Unload an executable from memory.
Control+V	Display Version Information window.
Control+K	Display Key window.
Control+C	Copy information to the clipboard.
Control+P	Print information/report.
Control+T	Set Filters for the main window.
F1	View Help.
F2	Check exported function calls in the selected module.
F3	Check links for the selected module.
F4	Check imported function calls in the selected module.
F5	Refresh main window.
F6	Display old executable (MZ) header.
F7	Display new executable (NE) header.
F8	Display portable executable (PE) header.
F9	Display memory status information.
Alt+F4	Quit Memory Check.

Key actions in the **Function Check** main window are as follows:

Tab	Switch between the main and the information windows.
Enter	View version information window.
Spacebar	Select a module.
Arrow keys	Navigate up/down the list of modules.
Control+O	Open executable for examination.
Control+V	Display Version Information window.
Control+K	Display Key window.
Control+C	Copy information to the clipboard.
Control+P	Print information/report.
Control+L	Load selected module into Function Check.
F1	View Help.
F2	Display exported functions.
F3	Check links to other modules.
F4	Display imported functions.
F5	Refresh main window.
F6	Display old executable (MZ) header.
F7	Display new executable (NE) header.
F8	Display portable executable (PE) header.
Alt+F4	Quit Function Check.

Menu Options

Please select one of the following:

Link Check Menu Options
Memory Check Menu Options
Function Check Menu Options

Link Check Menu Options

Please select one of the following:

File:
Open Executable
Open ODBC Data Source
Save As
Print
Run Executable
Run Executable with Parameters
Self-Register Executable
Unregister Executable
Exit

Options:
Copy
Copy Plain File Names
Display Version Information
Display Key to Symbols
Display Old Executable (MZ) Header
Display New Executable (NE) Header
Display Portable Executable (PE) Header
Check Exported Function Calls
Check Imported Function Calls
Check Modules in Memory
Expand All Levels
Expand One More Level
Collapse All

Preferences:
Show Modules Currently in Memory
Show Optional Modules
Show Self-Registering Executables
Ignore Modules in Memory
Show Lowercase File Names
Show Thick Lines
Bold Font
Show Toolbar Tips
Show Toolbar
Show Status Bar
Show Settings Now
Save Settings on Exit

Help:
Contents
Search for Help on
How to Use Help
About Link Check

Memory Check Menu Options

Please select one of the following:

File:
Load
Unload
Save As
Print
Exit

Options:
Copy
Display Version Information
Display Key to Symbols
Display Old Executable (MZ) Header
Display New Executable (NE) Header
Display Portable Executable (PE) Header
Check Exported Function Calls
Check Imported Function Calls
Check Links to Other Modules
Display Memory Status
Refresh
Set Filters

Preferences:
Show Lowercase File Names
Bold Font
Show Toolbar Tips
Show Toolbar
Show Status Bar
Show Settings Now
Save Settings on Exit

Help:
Contents
Search for Help on
How to Use Help
About Memory Check

Function Check Menu Options

Please select one of the following:

File:
Open Executable
Save As
Print
Exit

Options:
Copy
Display Version Information
Display Key to Symbols
Display Exported Functions
Display Imported Functions
Display Old Executable (MZ) Header
Display New Executable (NE) Header
Display Portable Executable (PE) Header
Check Links to Other Modules
Examine Selected Module
Highlight Possible Problems
Refresh

Sort:
Ordinals Ascending (1-9)
Ordinals Descending (9-1)
Module Names Ascending (A-Z)
Module Names Descending (Z-A)
Function Names Ascending (A-Z)
Function Names Descending (Z-A)
No Sort

Preferences:
Show Lowercase Function Names
Show Lowercase Module Names
Show Hexadecimal Values
Bold Font
Show Toolbar Tips
Show Toolbar
Show Status Bar
Show Settings Now
Save Settings on Exit

Help:
Contents
Search for Help on
How to Use Help
About Function Check

 File: Open Executable

Click on this menu option or on the short cut on the Toolbar to open an executable file for examination.

 File: Open ODBC Data Source

Click on this menu option to open an ODBC data source driver for examination.

The Open ODBC Data Source dialog will pop up for you to select the data source.

 File: Save As

Click on this menu option or click on the short cut on the Toolbar to save the information in the main window to a file.

 File: Print

Click on this menu option or click on the short cut on the Toolbar to print the information in the main window.

File: Run Executable

Click on this menu option to run the currently loaded executable file.

File: Run Executable with Parameters

Click on this menu option to run the currently loaded executable file with parameters.

The Run Executable with Parameters dialog will pop up for you to enter the parameters.

File: Self-Register Executable

Click on this menu option to make the executable write its settings into the Registry.

Some DLLs use OLE/COM settings which must be installed correctly in the system Registry for the module to function correctly. If you suspect the DLL is not behaving as expected, use this option to force the DLL to write its settings back to the Registry.

Link Check uses **REGSVR32.EXE** to write the values to the Registry.

This option is only valid in the 32-bit versions of Link Check.

File: Unregister Executable

Click on this menu option to make the executable remove its settings from the Registry.

You may use this option to remove settings, perhaps corrupt, from the Registry and then write them back using the **Self-Register Executable** menu option.

This option is only valid in the 32-bit versions of Link Check.

 File: Exit

Click on this menu option or click on the short cut on the Toolbar to exit the program.

File: Open Previous Module

Click on this menu option to re-open a recently used module.

File: Open Previous Module

Click on this menu option to re-open a recently used module.

File: Open Previous Module

Click on this menu option to re-open a recently used module.

File: Open Previous Module

Click on this menu option to re-open a recently used module.

 Options: Copy

Click on this menu option or click on the short cut on the Toolbar to copy information to the clipboard.

Click on the window you're interested in and the choose this option.

Options: Copy Plain File Names

Click on this menu option to copy the plain file names into the clipboard. The information can then be saved as a file and passed onto a zipping program to make a .ZIP file from the modules. Useful for backing up purposes.

Options: Display Version Information

Click on this menu option to show the **Version window** at the right-hand side of the main window.

This window will show version and file details of the selected module and can also be invoked from the popup menu by **right-clicking** with the mouse anywhere over the main window.

You may hide this window either by **right-clicking** on the mouse whilst over this window or by re-selecting the same menu option.

Options: Display Key to Symbols

Click on this menu option to show the **Key window** at the right-hand side of the main window.

This window will show an explanation to the symbols used in the main window and can also be invoked from the popup menu by **right-clicking** with the mouse anywhere over the main window.

You may hide this window either by **right-clicking** on the mouse whilst over this window or by re-selecting the same menu option.

Options: Display Old Executable (MZ) Header

Click on this menu option to show the **Old Executable Header** for the selected executable at the right-hand side of the main window.

The MS-DOS (old-style) executable-file header contains four distinct parts: a collection of header information (such as the signature word, the file size, and so on), a reserved section, a pointer to a Windows header (if one exists), and a stub program.

Applications must verify the Windows header for each executable-file header being tested, because a few applications have a different header style. MS-DOS uses the stub program to display a message if Windows has not been loaded when the user attempts to run a program.

For more information about the MS-DOS executable-file header, see the Microsoft MS-DOS Programmer's Reference (Redmond, Washington: Microsoft Press, 1991).

Options: Display New Executable (NE) Header

Click on this menu option to show the **New Executable Header** for the selected executable at the right-hand side of the main window.

The Windows (new-style) executable-file header contains information that the loader requires for segmented executable files. This information includes the linker version number, data specified by the linker, data specified by the resource compiler, tables of segment data, tables of resource data, and so on.

Here are the meanings for each field displayed:

File Name: Name of the file selected.

Magic Number: Specifies the signature word. The low byte contains "N" (4Eh) and the high byte contains "E" (45h).

Linker Major Version: Specifies the linker version number.

Linker Minor Version: Specifies the linker revision number.

Offset of Entry Table: Specifies the offset to the entry table (relative to the beginning of the header).

Length of Entry Table in Bytes: Specifies the length of the entry table, in bytes.

32-bit Checksum: 32-bit CRC of entire contents of file. These words are taken as 00 during the calculation.

Flags:

0 = The executable-file format is NOAUTODATA. An executable file with this format does not contain an automatic data segment.

1 = SINGLEDATA (Shared automatic data segment). The linker sets this bit if the executable-file format is SINGLEDATA. An executable file with this format contains one data segment. This bit is set if the file is a dynamic-link library (DLL).

2 = MULTIPLEDATA (Instanced automatic data segment) The linker sets this bit if the executable-file format is MULTIPLEDATA. An executable file with this format contains multiple data segments. This bit is set if the file is a Windows application.

2048 = The first segment in the executable file contains code that loads the application.

8192 = Errors detected at link time, module will not load.

32768 = Library module. The SS:SP information is invalid, CS:IP points to an initialization procedure that is called with AX equal to the module handle. This initialization procedure must perform a far return to the caller, with AX not equal to if zero to indicate success, or AX equal to zero to indicate failure to initialize. DS is set to the library's data segment if the SINGLEDATA flag is set. Otherwise, DS is set to the caller's data segment. A program or DLL can only contain dynamic links to executable files that have this library module flag set. One program cannot dynamic-link to another program.

Automatic Data Segment Number: Segment number of automatic data segment. This value is set to zero if SINGLEDATA and MULTIPLEDATA flag bits are clear, NOAUTODATA is indicated in the flags word. A Segment number is an index into the module's segment table. The first entry in the segment table is segment number 1.

Initial Heap Size: Initial size, in bytes, of dynamic heap added to the data segment. This value is zero if no initial local heap is allocated.

Initial Stack Size: Initial size, in bytes, of stack added to the data segment. This value is zero to indicate no initial stack allocation, or when SS is not equal to DS.

Initial CS:IP Setting: Specifies the segment:offset value of CS:IP.

Initial SS:SP Setting: Specifies the segment:offset value of SS:SP. The value specified in SS is an index to the module's segment table. The first entry in the segment table corresponds to segment number 1. If SS addresses the automatic data segment and SP is zero, SP is set to the address obtained by adding the size of the automatic data segment to the size of the stack.

Entries in Segment Table: Specifies the number of entries in the segment table.

Entries in Module Reference Table: Specifies the number of entries in the module-reference table.

Size of Non-Resident Name Table: Specifies the number of bytes in the nonresident-name table.

Offset of Segment Table: Specifies a relative offset from the beginning of the Windows header to the beginning of the segment table.

Offset of Resource Table: Specifies a relative offset from the beginning of the Windows header to the beginning of the resource table.

Offset of Resident Name Table: Specifies a relative offset from the beginning of the Windows header to the beginning of the resident-name table.

Offset of Module Reference Table: Specifies a relative offset from the beginning of the Windows header to the beginning of the module-reference table.

Offset of Imported Names Table: Specifies a relative offset from the beginning of the Windows header to the beginning of the imported-name table.

Offset of Non-Resident Names Table: Specifies a relative offset from the beginning of the file to the beginning of the nonresident-name table.

Count of Moveable Entries: Specifies the number of movable entry points.

Segment Alignment Shift Count: Specifies a shift count that is used to align the logical sector. This count is log2 of the segment sector size. It is typically 4, although the default count is 9. (This value corresponds to the /alignment [/a] linker switch. When the linker command line contains /a:16, the shift count is 4. When the linker command line contains /a:512, the shift count is 9.)

Number of Resource Segments: Specifies the number of resource segments.

Target Operating System: Specifies the target operating system, depending on which bits are set. A value of **0** means the operating system is unknown; a value of **2** means the target operating system is Microsoft Windows.

Other .EXE Flags: Specifies additional information about the executable file. It can be one or more of the following values:

2 = If this value is set, the executable file contains a Windows 2.x application that runs in version 3.x protected mode.

4 = If this value is set, the executable file contains a Windows 2.x application that supports proportional fonts.

8 = If this value is set, the executable file contains a fast-load area.

Fast Load Offset (in Sectors): Specifies the offset, in sectors, to thebeginning of the fast-load area. (Only Windows uses this value.)

Fast Load Length (in Sectors): Specifies the length, in sectors, of thefast-load area. (Only Windows uses this value.)

Expected Windows Version Number: Specifies the expected version numberfor Windows. (Only Windows uses this value.)

Segment Sector Size: Specifies the segment sector size. This value is 1 shifted to the left by the Segment Alignment Shift Count.

Options: Display Portable Executable (PE) Header

Click on this menu option to show the **Portable Executable Header** for the selected executable at the right-hand side of the main window.

The PE/COFF (Common Object File Format) file headers consist of an MS-DOS stub, file signature, COFF Header, and Optional Header. An object file usually contains only the COFF Header, but an image file contains all the headers. In both cases, the file headers are followed immediately by section headers.

Here are the meanings for each field displayed:

File Name: Name of the file selected.

Magic Number: There is a 4-byte signature identifying the file as a PE format image file; this format is used in Win32, Posix on NT, and for some device drivers in Windows NT. Currently, this signature is "PE" (the letters "P" and "E" followed by two null bytes).

Machine: Number identifying type of target machine. Valid values are:

0 = Contents assumed to be applicable to any machine type.

332 = Intel 386 or later, and compatible processors.

350 = MIPS(r) little endian.

388 = Alpha AXP(tm).

496 = Power PC, little endian.

616 = Motorola 68000 series.

656 = Precision Architecture (PA) RISC processor.

Number of Sections: Number of sections; indicates size of the Section Table, which immediately follows the headers.

Time/Date Stamp: Time and date the file was created.

Pointer to Symbol Table: Offset, within the COFF file, of the symbol table.

Number of Symbols: Number of entries in the symbol table. This data can be used in locating the string table, which immediately follows the symbol table.

Size of Optional Header: Size of the optional header, which is included for executable files but not object files. An object file should have a value of 0 here.

Characteristics: Flags indicating attributes of the file. Valid values are:

1 = Image only. Indicates that the file does not contain base relocations and must therefore be loaded at its preferred base address. If the base address is not available, the loader reports an error. Operating systems running on top of MS-DOS (Win32s) are generally not able to use the preferred base address and so cannot run these images. However, beginning

with version 4.0, Windows will use an application's preferred base address.

2 = Image only. Indicates that the image file is valid and can be run. If this flag is not set, it generally indicates a linker error.

4 = COFF line numbers have been removed.

8 = COFF symbol table entries for local symbols have been removed.

16 = Reserved for future use.

32 = Reserved for future use.

64 = Use of this flag is reserved for future use.

128 = Little endian: LSB precedes MSB in memory.

256 = Machine based on 32-bit-word architecture.

512 = Debugging information removed from image file.

1024 = Reserved for future use.

4096 = The image file is a system file, not a user program.

8192 = The image file is a dynamic-link library (DLL). Such files are considered executable files for almost all purposes, although they cannot be directly run.

327680 = Big endian: MSB precedes LSB in memory.

Optional Header Signature: Unsigned integer identifying the state of the image file. The most common number is 0413 octal (0x10B), identifying it as a normal executable file. 0407 (0x107) identifies a ROM image.

Linker Major Version: Linker major version number.

Linker Minor Version: Linker minor version number.

Size of Code: Size of the code (text) section, or the sum of all code sections if there are multiple sections.

Size of Initialised Data: Size of the initialized data section, or the sum of all such sections if there are multiple data sections.

Size of Uninitialised Data: Size of the uninitialized data section (BSS), or the sum of all such sections if there are multiple BSS sections.

Address of Entry Point: Address of entry point, relative to image base, when executable file is loaded into memory. For program images, this is the starting address. For device drivers, this is the address of the initialization function.

Base of Code: Address, realtive to image base, of beginning of code section, when loaded into memory.

Base of Data: Address, realtive to image base, of beginning of data section, when loaded

into memory.

Image Base: Preferred address of first byte of image when loaded into memory; must be a multiple of 64K.

Section Alignment: Alignment (in bytes) of sections when loaded into memory. Must greater or equal to File Alignment. Default is the page size for the architecture.

File Alignment: Alignment factor (in bytes) used to align pages in image file. The value should be a power of 2 between 512 and 64K inclusive.

Major Operating System Version: Major version number of required OS.

Minor Operating System Version: Minor version number of required OS.

Major Image Version: Major version number of image.

Minor Image Version: Minor version number of image.

Major Subsystem Version: Major version number of subsystem.

Minor Subsystem Version: Minor version number of subsystem.

Size of Image: Size, in bytes, of image, including all headers; must be a multiple of Section Alignment.

Size of Headers: Combined size of MS-DOS Header, PE Header, and Object Table.

Checksum: Image file checksum. The algorithm for computing is incorporated into IMAGHELP.DLL. The following are checked for validation at load time: all drivers, any DLL loaded at boot time, and any DLL that ends up in the server.

Subsystem: Subsystem required to run this image. Valid values are:

0 = Unknown subsystem.

1 = Used for device drivers and native Windows NT processes.

2 = Image runs in the Windows graphical user interface (GUI) subsystem.

3 = Image runs in the Windows character subsystem.

7 = Image runs in the Posix character subsystem.

DLL Characteristics: Obsolete.

Size of Stack Reserve: Size of stack to reserve. Only the Stack Commit Size is committed; the rest is made available one page at a time, until reserve size is reached.

Size of Stack Commit: Size of stack to commit.

Size of Heap Reserve: Size of local heap space to reserve. Only the Heap Commit Size is committed; the rest is made available one page at a time, until reserve size is reached.

Size of Heap Commit: Size of local heap space to commit.

Loader Flags: Obsolete.

Number of Data Directories: Number of data-dictionary entries in the remainder of the Optional Header. Each describes a location and size.

 Options: Check Exported Function Calls

Click on this menu option or on the short cut on the Toolbar to launch **Function Check** component of **Link Check**.

You can examine all of the function calls in the module as well as the ones that it requires to be present in the system to run properly. Function Check will start in the **Exported** view.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F2**.

 Options: Check Imported Function Calls

Click on this menu option or on the short cut on the Toolbar to launch **Function Check** component of **Link Check**.

You can examine all of the function calls in the module as well as the ones that it requires to be present in the system to run properly. Function Check will start in the **Imported** view.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F4**.

 Options: Check All Modules in Memory

Click on this menu option or on the short cut on the Toolbar to launch **Memory Check** component of **Link Check**.

You can examine all of the modules currently loaded in memory.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F3**.

Options: Expand All Levels

Click on this menu option to show every single module required by the executable.

Options: Expand One More Level

Click on this menu option to expand the current level throughout the Main Window.

When a module is first opened, the first level of linked modules are shown. A second level of modules can be seen if any one of the first level modules are expanded by **double-clicking** on the module, and so on for up to 256 levels.

Rather than having to select each first level module for expansion, you can select just one and use this menu option to expand all of the modules on the same level as the selected one. The same holds true for any level.

Note that only the modules on the same level and underneath get expanded, so select the topmost module if you want to see the whole level expanded.

Options: Collapse All

Click on this menu option to collapse all open links of the loaded module.

This menu option is equivalent to re-opening the executable file.

Preferences: Show Modules Currently in Memory

Click on this menu option to show the status of each module whether loaded in memory or not.

If a  symbol appears next to the module name then the module is loaded in memory.

If a  symbol appears next to the module name then the module is **not** loaded in memory.

Preferences: Show Optional Modules

Click on this menu option to make **Link Check** scan the currently open executable for the string **.DLL** to find out if there are any references to some external .DLLs embedded in the code.

Any entries found are shown in Blue and these modules are **not** necessarily required by the system for the executable to work properly.

Preferences: Show Self-Registering Executables

Click on this menu option to tell **Link Check** to display modules which are self-registering.

Self-registering modules are those that write their settings into the system Registry and without which they could not function properly.

You may use the **Self-Register Executable** menu option to force a module to write its settings to the Registry in case they become corrupt.

This option is valid only in the 32-bit versions of Link Check.

Preferences: Ignore Modules in Memory

Click on this menu option to tell **Link Check** to ignore modules in memory when resolving links to modules.

This will give a picture of what modules **would be** loaded if there wasn't any programs running (and possibly loading their own versions of modules not necessarily present in the target machine).

Preferences: Show Lowercase File Names

Click on this menu option to draw the file names in the main window in lowercase or as is.

If this menu option is checked, then the file names will be in lowercase, otherwise as is.

Preferences: Show Thick Lines

Click on this menu option to draw the connecting modules with either thick or thin lines.

If this menu option is checked, then the lines will be thick, otherwise thin.

Preferences: Bold Font

Click on this menu option to turn bold font on and off.

If this menu option is checked, then the font will be bold, otherwise normal.

Preferences: Show Toolbar Tips

Use this menu option to turn the Toolbar button explanations/hints on and off.

Preferences: Show Toolbar

Use this menu option to show or hide the Toolbar.

Preferences: Show Status Bar

Use this menu option to show or hide the Status Bar.

Preferences: Save Settings Now

Select this menu option if you want the program to save the current position and size of the main window plus all other settings now.

When you restart the program, it will reposition, size and configure itself to the saved values.

Preferences: Save Settings on Exit

Select this menu option if you want the program to save the current position and size of the main window plus all other settings when you exit the program.

When you restart the program, it will reposition, size and configure itself to the saved values.

Help: Contents

This menu item shows the Contents page of the help file. Alternatively, click on the short cut on the Toolbar.

Help: Search for Help on

This menu item shows the Search dialog box for the help file.

Help: How to Use Help

This menu item shows how to use Help.

Help: About

Use this dialog box to view the copyright information for this program.

 File: Load

Click on this menu option or on the short cut on the Toolbar to load an executable into memory.

 File: Unload

Click on this menu option or on the short cut on the Toolbar to unload an executable from memory.

WARNING: Unloading a module from memory can cause damage to your system. This option should be used with the utmost of care and is intended for use only when a program has ended abnormally and has left module(s) lying around in memory.

Karri Software Limited is in **NO WAY RESPONSIBLE for any damage done** and **the user must assume the FULL RISK** on using it.

You will be warned before this action is executed and you will need to confirm this action.

 Options: Check Links to Other Modules

Click on this menu option or on the short cut on the Toolbar to launch **Link Check**.

You can view a list of all of the modules that the selected one is linked to in the system.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F3**.

Options: Memory Status

Click on this menu option to display **Memory Status** information on theright-hand side of the main window.

For the 16-bit version the information displayed is as follows:

The **Free System Resources** field displays the percentage of free system resources.

The **Free GDI Resources** field displays the percentage of free GDI.EXE resources.

The **Free User Resources** field displays the percentage of free USER.EXE resources.

The **Largest Free Block** field displays the largest free block of contiguous linear memory in the system, in bytes.

The **Max. Pages Available** field specifies the maximum number of pages that could be allocated in the system (the value of the Largest Free Block divided by the value of Page Size).

The **Max. Pages Lockable** field specifies the maximum number of pages that could be allocated and locked.

The **Total Linear Space** field specifies the size of the total linear address space, in pages.

The **Total Unlocked Pages** field specifies the number of unlocked pages in the system. This value includes free pages.

The **Free Pages** field specifies the number of pages that are not in use.

The **Total Pages** field specifies the total number of pages the virtual-memory manager manages. This value includes free, locked, and unlocked pages.

The **Free Linear Space** field specifies the amount of free memory in the linear address space, in pages.

The **Swap File Pages** field specifies the number of pages in the system swap file.

The **Page Size** field specifies the system page size, in bytes.

The **Free Memory** field specifies the amount of free memory, in bytes. This is equivalent to what Program Manager reports via its About dialog.

For the 32-bit versions the information displayed is as follows:

The **Memory Load** field specifies a number between 0 and 100 that gives a general idea of current memory utilization, in which 0 indicates no memory use and 100 indicates full memory use.

The **Total Physical Memory** field indicates the total number of bytes of physical memory.

The **Free Physical Memory** field indicates the number of bytes of physical memory available.

The **Total Page File** field indicates the total number of bytes that can be stored in the paging file. Note that this number does not represent the actual physical size of the paging file on disk.

The **Free Page File** field indicates the number of bytes available in the paging file.

The **Total Virtual Memory** field indicates the total number of bytes that can be described in the user mode portion of the virtual address space of the calling process.

The **Free Virtual Memory** field indicates the number of bytes of unreserved and uncommitted memory in the user mode portion of the virtual address space of the calling process.

Click on the window for instant refresh.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F6**.

Options: Refresh

Click on this menu option to refresh the main window with updated information.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F5**.

 Options: Set Filters

Click on this menu option to display the Set Filters dialog box.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **Control+T**.

Options: Display Exported Functions

Click on this menu option to view all of the exported function names for the module.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F2**.

An **Exported Function** is a code routine written to be used by other external programs or modules. These functions usually reside in .DLLs (Dynamic Link Libraries) and are **Imported** by programs (.EXEs) as well as by other modules (such as other .DLLs).

Functions are exported by name and by ordinal. An ordinal is an index to the function.

Options: Display Imported Functions

Click on this menu option to view all of the imported function names for the module.

This menu option can also be invoked by **right-clicking** with the mouse anywhere in the main window or by pressing **F4**.

An **Imported Function** is a code routine written in an external module (such as a .DLL) which the calling program or module uses as part of itself. The external module **Exports** the function so that it is available for all to use.

Problems can occur when a program is using a newer version of a .DLL, say, and the .DLL gets overwritten by an older one which does not include the function imported by the program. In this case the system would display a **"Call to Undefined Dynalink"** error message.

Functions can be imported by name or by ordinal. An ordinal is an index to the function.

Please see Highlight Possible Problems menu option for more information on how to spot these potential errors.

Options: Examine Selected Module

Click on this menu option to launch a second copy of **Function Check** to view functions in the selected module.

This option is only available when viewing **Imported** function names.

Options: Highlight Possible Problems

Click on this menu option to view possible problem entries in the executable module.

These entries are shown by the  symbol and upon selecting this option, only these entries are shown in the window.

Please note that this option is only available when viewing **Imported** function names.

 Sort: Ordinals Ascending (1-9)

Click on this menu option to sort the ordinals in ascending order.

 Sort: Ordinals Descending (9-1)

Click on this menu option to sort the ordinals in descending order.

 Sort: Module Names Ascending (A-Z)

Click on this menu option to sort the module names in ascending order.

 Sort: Module Names Descending (Z-A)

Click on this menu option to sort the module names in descending order.

 Sort: Function Names Ascending (A-Z)

Click on this menu option to sort the function names in ascending order.

 Sort: Function Names Descending (Z-A)

Click on this menu option to sort the function names in descending order.

Sort: No Sort

Preferences: Show Lowercase Function Names

Click on this menu option to draw the function names in the main window in lowercase or as is.

If this menu option is checked, then the function names will be in lowercase, otherwise as is.

Preferences: Show Lowercase Module Names

Click on this menu option to draw the module names in the main window in lowercase or as is.

If this menu option is checked, then the module names will be in lowercase, otherwise as is.

Preferences: Show Hexadecimal Values

Click on this menu option to display the Ordinals (and Hints in 32-bit versions) in either hexadecimal or decimal.

If this menu option is checked, then the ordinals will be in hexadecimal, otherwise decimal.

Toolbars

The **Link Check** Toolbar contains the following short cuts:

Open Executable

Open ODBC Data Source

Save As

Print

Copy

Check Exported Function Calls

Check Imported Function Calls

Check All Modules in Memory

Exit Program

Help Contents

The **Memory Check** Toolbar contains the following short cuts:

Load

Unload

Save As

Print

 Copy

 Check Exported Function Calls

 Check Imported Function Calls

 Check Links to Other Modules

 Set Filters

 Exit Program

 Help Contents

The **Function Check** Toolbar contains the following short cuts:

 Open Executable

 Save As

 Print

 Copy

 Check Exported Function Calls

 Check Imported Function Calls

 Ordinals Ascending (1-9)

 Ordinals Descending (9-1)

 Module Names Ascending (A-Z)

 Module Names Descending (Z-A)

 Function Names Ascending (A-Z)

 Function Names Descending (Z-A)

 Exit Program

 Help Contents

Dialog Boxes

Link Check Dialog Boxes:

Open ODBC Data Source dialog
Run Executable with Parameters dialog

Memory Check Dialog Boxes:

Set Filters dialog

Function Check does not have any Dialog Boxes.

Open ODBC Data Source Dialog

This dialog box is invoked from the **Open ODBC Data Source** menu option under the **File** menu.

In this dialog, select an **ODBC*** (Open Database Connectivity) data source for examination and click on **OK**.

For this dialog box to appear you need ODBC installed on your computer.

Link Check will find the appropriate ODBC driver for the data source and display the required modules.

Click on **Cancel** to return to the main window without any further action.

* See Trademarks

Run Executable with Parameters Dialog

This dialog box is invoked from the **Run Executable with Parameters** menu option under the **File** menu.

In this dialog, enter any extra parameters that you would like to pass on to the program about to be executed and click on **OK**.

Do not include the program name in the field, only the parameters.

Link Check will precede the parameters with the program name before attempting to execute it.

Click on **Cancel** to return to the main window without any further action.

Set Filters Dialog

This dialog box is invoked from the **Set Filters** menu option under the **Options** menu.

In this dialog, enter the file extensions that you would only like to see in the main window.

Separate each extension with a semi-colon (;). You may include the dot (.) in the extension even though it's not necessary.

Example: **DLL;EXE;FON**

Leave the entry field **blank** to view **all** modules in memory.

Click on **OK** to view only the modules which have the specified extensions.

Click on **Cancel** to return to the main window without any further action.

Registered Version Details

Link Check is shareware and not available for free. You are allowed a 30-day evaluation period free of charge for this program and then you should either register it with the author or remove it from your disk(s). Should you use it after the 30-day evaluation period, you are violating US and international copyright laws and treaties.

Each registered version of **Link Check** includes:

1. All references to "Unregistered Shareware" removed
2. Your name and organisation in the "About" dialog box

As a registered user of **Link Check**, you will receive the following benefits:

1. Free technical support via Email
2. Free upgrades via the license file

Currently (Oct 97) we have no plans to charge for upgrades. Once the license file is installed, any upgrades will be seamless. All you need to do is to obtain/download a new copy and install over the old version. The presence of the license file will unlock the retail version automatically so you need to register **only once**.

Copyright and License Agreement

Link Check 5.70
Memory Check 5.70
Function Check 5.70
Copyright (C) Karri Software Limited 1994-1997

The above is from now on known as SOFTWARE.

Shareware License

You should read this section carefully **before** using this SOFTWARE. By using this SOFTWARE you are agreeing to the terms and conditions of this license agreement.

This SOFTWARE is shareware and not available for free. You are hereby granted a license for a 30-day evaluation period free of charge for this SOFTWARE. After the evaluation period has passed you **must** either register the SOFTWARE with Karri Software Limited or remove it from your disk(s). Should you use the SOFTWARE after the 30-day evaluation period, you are violating US and International copyright laws and treaties.

You are hereby granted a license to make as many copies of the SOFTWARE as you wish, and distribute it to anyone as long as the SOFTWARE is distributed in its entirety and without any modifications what so ever. Bulletin Board Systems are allowed to add a small file describing the BBS with the SOFTWARE.

You are prohibited from charging or requesting donations from this SOFTWARE in any form. You are also prohibited in distributing the SOFTWARE as part of other products (commercial or otherwise) without prior written permission from Karri Software Limited. Only Disk Vendors approved by the Association of Shareware Professionals are permitted to redistribute the SOFTWARE, subject to the conditions in this license agreement, without prior written permission.

LIMITATIONS:

You may not rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer or transfer the licensed SOFTWARE except as provided in this agreement. Any such unauthorised use shall terminate this license immediately.

All rights not expressly granted here are reserved to Karri Software Limited.

Retail License

GRANT OF LICENSE: This license grants you the following rights:

You are hereby granted one license of the SOFTWARE. If you are a single entity, you may designate one individual within your organization to have the right to use the SOFTWARE in the manner described in this license agreement.

You may make one (1) copy of the SOFTWARE solely for backup or archiving purposes.

You may also store or install one copy of the SOFTWARE on a storage device, such as a

network server, used only to install or run the SOFTWARE on your other computers over an internal network; however, you must acquire and dedicate a license for each separate computer on which the SOFTWARE is installed or run from the device storage. A license cannot be shared or used concurrently on different computers.

LIMITATIONS:

You may not rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer or transfer the licensed SOFTWARE except as provided in this agreement. Any such unauthorised use shall terminate this license immediately.

All rights not expressly granted here are reserved to Karri Software Limited.

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

THE USER MUST ASSUME THE FULL RISK OF USING THIS SOFTWARE.

ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF THE PURCHASE PRICE WITHIN 90 DAYS OF RECEIPT.

Trademarks

Windows is a registered trademark of Microsoft Corporation.
Windows 3.1, Windows for Workgroups 3.11, Windows 95 and Windows NT are registered trademarks of Microsoft Corporation.
OS/2 is a registered trademark of International Business Machines Corporation.
Visual Basic is a trademark of Microsoft Corporation.
ODBC is a trademark of Microsoft Corporation.

Official Home

The official home of **Link Check** is the **Windows Utility (GO WINUTIL)** forum on CompuServe. You can direct technical questions about **Link Check** to the support staff in this forum.