

User's Guide

ISeamless 1.2.0



2805 Bowers Avenue
Santa Clara, CA 95051

Phone: (408) 988-3832
Monday - Friday
6:00 A.M. - 6:00 P.M.

FAX: (408) 970-9727
BBS: (408) 988-4004

COPYRIGHT

Copyright © 1997, 1998 by Network Associates, Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Network Associates, Inc.

TRADEMARK NOTICES

McAfee, Network Associates, VirusScan, NetShield, and Site Meter are registered trademarks of Network Associates, Inc. WebScan, WebScanX, SiteExpress, BootShield, ServerStor, ScreenScan, WebCrypto, PCCrypto, PCFirewall, NetCrypto, GroupShield, GroupScan, Remote Desktop 32, WebShield, NetRemote, eMail-It, Hunter, ScanPM, and SecureCast are trademarks of Network Associates, Inc. All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations.

“SABRE” is a trademark of American Airlines, Inc. and is licensed for use to Network Associates. Saber Software is not affiliated with American Airlines, Inc. or SABRE Travel Information Network. All trademarks are the property of their respective owners.

FEEDBACK

Network Associates appreciates your comments and reserves the right to use any information you supply in any way it believes appropriate without incurring any obligations whatsoever. Please address your feedback to: Network Associates, Inc., Documentation, 2805 Bowers Avenue, Santa Clara, CA 95051-0963, send e-mail to documentation@nai.com, or send a fax to Documentation at (408) 970-9727.

Table of Contents

Chapter 1. Introducing ISeamless.....	5
What Is ISeamless?	5
What Is a Script?	6
How To Contact Network Associates	7
Customer service	7
Technical support.....	7
Network Associates training.....	8
International contact information.....	9
Chapter 2. Using ISeamless	11
System Requirements	11
ISeamless Quick Start	11
Chapter 3. Writing a Script	13
General Syntax Information	15
Chapter 4. Understanding ISeamless	16
Specifying Timing.....	16
Specifying Actions	17
Specifying Directory Locations	18
Setup Command-line Options.....	20
SISSRC and SISABS.....	20
Chapter 5. Examining Three Examples	22

Chapter 6. Defining Actions24

CopyFile.....	25
DeleteFile.....	27
RunCustomProgram	28
SetRegValue	29
SetIniValue	31
FileAttribute.....	33
EnterDisk	35
DeleteRegValue	36
DeleteRegKey.....	38
AddGroupIcon.....	39
DeleteGroupIcon.....	41
DeleteGroup	42
VariableValue	43
NCCONFIGDEBUG	43
NNEEDREBOOT	44
ISSILENT_FINISHDELAY.....	45
ISSILENT_ABNORMAL.....	46

Chapter 7. Using the Compiler47

Appendix A. Sample Scripts.....50

Using a Custom Licensing Agreement	50
Using Custom .VSC and .VSH Files.....	51
Using Newer .DAT Files.....	52

Appendix B. Compilation Error Codes53

Index55

Introducing ISeamless

What Is ISeamless?

ISeamless scripting is a tool that network administrators and OEM vendors can use to customize software installation to fit the needs of their clients. The scripting language enables convenient site-wide deployment of VirusScan for Windows 95 version 3.0.1 or later, VirusScan for Windows NT, VirusScan for Windows 3.x, NetShield for Windows NT, and GroupShield. Here are some examples of tasks that you can accomplish with ISeamless:

- Example 1: An OEM vendor wants to install a version of anti-virus or network security software that displays a custom licensing agreement during setup. The installation script must overwrite the shipping license agreement with the custom one.
- Example 2: An administrator wants to deploy a version of VirusScan with customized versions of the DEFAULT.VSC and DEFAULT.VSH files, and to set a ScreenScan registry key so that ScreenScan scans only program files instead of all files.
- Example 3: An administrator wants to distribute VirusScan with a newer set of .DAT files from Network Associates.

Although solutions for some of these problems already existed, they were neither very clean nor very reliable. The licensing issue, in particular, did not have any solution before. Using ISeamless scripting makes all of these example scenarios not only possible, but also easier to implement.

What Is a Script?

A *script* is a list of commands (collection of actions) that can be executed without user interaction. A script can also be thought of as a batch file or macro. A *scripting language* is a simple programming language with which you can write scripts.

You can use ISeamless to write custom installation instructions for your software configuration files. The software installation executes these custom actions, such as copying a licensing agreement to the hard drive and displaying it, or setting options in the registry. The order of the specific actions can be broken up into phases or sections. Using this manual, you can explore the predefined set of phases, actions, and variables needed for optimum usage of ISeamless scripting.

How To Contact Network Associates

Customer service

To order products or obtain product information, we invite you to contact our Customer Care department by calling (408) 988-3832 or by writing to the following address:

Network Associates, Inc.
2805 Bowers Avenue
Santa Clara, CA 95051-0963
U.S.A.

Technical support

Network Associates is famous for its dedication to customer satisfaction. We have continued this tradition by making our site on the World Wide Web a valuable resource for answers to technical support issues. We encourage you to make this your first stop for answers to frequently asked questions, for updates to Network Associates software, and for access to Network Associates news and virus information.

World Wide Web

<http://www.nai.com>

If you do not find what you need or do not have web access, try one of our automated services.

Automated Voice and Fax
Response System

(408) 988-3034

Internet

support@nai.com

BBS

(408) 988-4004

1200 bps to 28,800 bps

8 bits, no parity, 1 stop bit

24 hours, 365 days a year

CompuServe

GO MCAFEE

America Online

keyword MCAFEE

If the automated services do not have the answers you need, contact Network Associates at one of the following numbers Monday through Friday between 6:00 A.M. and 6:00 P.M. Pacific time.

For corporate-licensed customers:

Phone (408) 988-3832

Fax (408) 970-9727

For retail-licensed customers:

Phone (972) 278-6100

Fax (408) 970-9727

To provide the answers you need quickly and efficiently, the Network Associates technical support staff needs some information about your computer and your software. Please have this information ready before you call:

- Product name and version number
- Computer brand and model
- Any additional hardware or peripherals connected to your computer
- Operating system type and version numbers
- Network type and version, if applicable
- Contents of your AUTOEXEC.BAT, CONFIG.SYS, and system LOGIN script
- Specific steps to reproduce the problem

Network Associates training

For information about scheduling on-site training for any Network Associates product, call (800) 338-8754.

International contact information

To contact Network Associates outside the United States, use the addresses and numbers below.

Network Associates Canada

139 Main Street, Suite 201
Unionville, Ontario
Canada L3R 2G6
Phone: (905) 479-4189
Fax: (905) 479-4540

Network Associates France S.A.

50 rue de Londres
75008 Paris
France
Phone: 33 1 44 908 737
Fax: 33 1 45 227 554

Network Associates (UK) Ltd.

Hayley House, London Road
Bracknell, Berkshire
RG12 2TH
United Kingdom
Phone: 44 1344 304 730
Fax: 44 1344 306 902

Network Associates Europe B.V.

Gatwickstraat 25
1043 GL Amsterdam
The Netherlands
Phone: 31 20 586 6100
Fax: 31 20 586 6101

Network Associates Deutschland GmbH

Industriestrasse 1
D-82110 Germering
Germany
Phone: 49 8989 43 5600
Fax: 49 8989 43 5699

Network Associates Japan Co, Ltd.

Toranomon 33 Mori Bldg.
3-8-21 Toranomon
Minato-Ku, Tokyo 105
Japan
Phone: 81 3 5408 0700
Fax: 81 3 5408 0780

Network Associates Korea

135-090, 18th Fl., Kyoung Am Bldg.
157-27 Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82 2 555-6818
Fax: 82 2 555-5779

Network Associates Australia

Level 1, 500 Pacific Highway
St. Leonards, NSW 2065
Australia

Network Associates South East Asia

7 Temasek Boulevard
The Penthouse
#44-01, Suntec Tower One
Singapore 038987
Tel: 65 430-6670
Fax: 65 430-6671

Network Associates Latin America

150 South Pine Island Road, Suite 205
Plantation, FL 33324
USA

System Requirements

- The ISeamless compiler runs on Windows 95 or Windows NT.
- ISeamless scripts run on Windows 3.x, Windows 95, and Windows NT.

ISeamless Quick Start


Here's an overview of basic steps to create an .ISC text file, compile it, and then use it with your software's SETUP.EXE:

1. Open an ASCII text editor.
2. Perform the following for all sections you want to perform actions in:
 - Enter the SectionName(), followed by an open brace "{".
 - Enter one or more actions you want to perform.
 - Enter a close brace "}".
3. When the file is complete, save it (with an .ISC extension) to the directory containing ISEAMCMP.EXE.
4. Run the compiler:
 - From your Windows 95 or Windows NT Start menu, point to Programs, then click MS-DOS Prompt.


- Change to the directory that contains ISEAMCMP.EXE.
- Type:

```
ISEAMCMP FILENAME.ISC NEWNAME.SIS
```

Response: A new file called NEWNAME.SIS appears in the current directory.

 *If compilation fails, error messages appear. For a list of possible error messages and their definitions, see [Appendix B, "Compilation Error Codes."](#) For brief descriptions of the commands and variables that can be used in the source file, see [Chapter 4, "Understanding ISeamless."](#) For detailed definitions of ISeamless commands and variables, see [Chapter 6, "Defining Actions."](#)*

5. Rename the file with a valid .SIS filename. Valid filenames for automatic installs are ADMIN.SIS, OEM.SIS, and AUTO*.SIS (any file starting with AUTO and having .SIS as its extension).

 *If you name the .SIS file with an invalid filename for automatic installs, you must use SISSRC or SISABS as the first parameter on your setup command line.*


6. Copy or move the newly compiled .SIS file to the source directory containing your software's SETUP.EXE and other installation files.
7. Copy any additional files needed by your script.
8. Run SETUP.EXE, in silent or non-silent mode.

The ISeamless compiler is based on C programming code and follows its function and construct, however, it only supports about 5 percent of the C language. If you are familiar with C code, writing ISeamless scripts will be easy, but beware of using C commands not listed in this document because most will not work.

The compiler does not have a built-in text editor, so open any ASCII text editor in either DOS or Windows. Save the file in an MS-DOS text format with an .ISC extension. In truth, the script file can have any extension, but for organizational purposes Network Associates recommends using an .ISC extension.

Network Associates upgrades its data files approximately every four to six weeks, but only prints VirusScan for Windows 95, for example, to diskette or CD once each quarter. Therefore, you only have a four-week window when the data files on the diskette or CD are current. One of the best features of ISeamless is its ability to help you build a script for deploying your newly received version of anti-virus or network security software with the latest set of data files available on the Network Associates website:

```
PostCopyFile( )
{
    CopyFile( "%ISSRCDIR%Dats.zip" ,
              "*.DAT" ,
              " " ,
              " " ,
              "%ISSVTARGET%" ,
              0 );
}
```

 *For the purposes of instruction and to make troubleshooting easier, each command parameter is put on a separate line. When writing your own scripts, you can put all of your command parameters on the same line.*

Here's an explanation of the above .DAT file update script:

- Line 1 contains the subject header, **PostCopyFile()**.

The subject header describes a section or phase of the install. The parentheses identify PostCopyFile as a function. C is a modular language built with functions. When the compiler reads a section in the script, it applies the statement that follows in the named phase of the installation.

- In Line 2, the opening brace { indicates the beginning of the statements that make up the function.
- Line 3 begins with the command **CopyFile** followed by an open parenthesis (to separate the command from its parameters.
- CopyFile has six parameters: Five are STRINGS, and one is a number.
 - Always end-cap STRING parameters with quotation marks (" ").
 - Use a comma (,) to separate parameters.
- The compiler recognizes the "%ISSRCDIR%" variable to mean the directory where the installation files are stored.

Always end-cap variables with percent symbols (% %).

- A number does not need quotation marks when it is interpreted as an integer and not a STRING. However, when a number is a STRING, end-cap it with quotation marks.
- You must use a close parenthesis) to indicate the end of the parameters.

General Syntax Information

The compiler is not case-sensitive. However, the examples here present commands and variables with mixed-case letters so that you can understand the commands better, or more easily differentiate between a command and a variable. When writing your own script, you can use either single- or mixed-case characters.

- To use the backslash character in a string, you must type two consecutive backslashes (`\\`). The ISeamless compiler interprets two backslashes between quotation marks as a single backslash (`\`). For example, `C:\` would be represented in the install script between quotations as `"C:\\"`.
- You can use long, descriptive filenames; the compiler recognizes them.
- You can add comments to your script in two ways: Type a semicolon followed by a space, or type `REM` followed by a space to remark out the line. To keep a command from running, or to make a comment about what the next line will do, either begin the line with two forward slashes (`//`), or end-cap the line or section with asterisks `/* */`.

Sample syntax:

```
// This is a comment
```

OR

```
/* This is also a comment */
```

OR

```
/* And this is a comment  
with a carriage return */
```

- You can use carriage returns and spaces. White space does not matter to the compiler; it looks for section names, braces, commands, parentheses, parameters, commas, quotation marks, and semicolons. If the syntax of the commands and their parameters is correct, you can put them all on the same command line without spaces.

4

Understanding ISeamless

Specifying Timing

Use the following section headers to specify in what section of the install subsequent commands execute:

Section Header	When Commands Execute
PreInitial()	Before the install sets its internal settings
PostInitial()	After the install sets its internal settings
PreLicense()	Before the licensing agreement appears
PostLicense()	After the licensing agreement appears
PreGetDest()	Before the install asks for a destination directory
PostGetDest()	After the install asks for a destination directory
PreCopyFile()	Before the install copies program files
PostCopyFile()	After the install copies program files
PreProductConfig()	Before the install configures .INI and registry settings
PostProductConfig()	After the install configures .INI and registry settings
PreCreatelcons()	Before the install creates program icons
PostCreatelcons()	After the install creates program icons
PreWhatsNew()	Before the install displays the What's New .TXT
PostWhatsNew()	After the install displays the What's New .TXT
PreFinish()	Before the install cleans up prior to shutting down (This is the last section executed if the install reboots.)
PostFinish()	After the install cleans up prior to shutting down (This section is NOT executed if the install reboots.)

Before or *after* implies “just before” or “just after.” Thus, a command in a “Pre-” section executes just before the phase itself. A command in a “Post-” section executes just after the last setup instruction of that phase. To execute actions in a section, type:

```
SectionName( )
{
    Action(...parameters...);
    Action(...parameters...);
    Action(...parameters...);
    ...
    Action(...parameters...);
}
```

Specifying Actions

Use the following commands to specify which actions to perform in a section:

Command	Description
CopyFile	Copies a file or set of files from a directory or a .ZIP archive to a target directory
DeleteFile	Deletes a file or a set of files from a directory
RunCustomProgram	Launches a program during the install
SetRegValue	Sets registry values
SetIniValue	Sets values in .INI files
FileAttribute	Sets the attributes of files
EnterDisk	Asks the user to enter a specific disk
DeleteRegValue	Deletes a value in the registry
DeleteRegKey	Deletes a whole key in the registry
AddGroupIcon	Adds an icon to a program group
DeleteGroupIcon	Deletes an icon from a program group
DeleteGroup	Deletes a group
VariableValue	Sets the value of installation internal variables

Specifying Directory Locations

The ISeamless compiler recognizes a group of preset variables. However, if you prefer, you can instead Copy and Paste, or rewrite your full path statements. To use the variables, you must end-cap them each with percent symbols (% %). These variables search for certain types of information; you don't need to configure them. For example, "ISWINSYSDIR" specifically looks for and identifies your Windows System directory, wherever it is located.

Use the following variables to specify directory locations in your script:

Variable	Directory or Disk Location	Example
ISWINSYSDISK	Disk containing the Windows System directory	C:
ISWINSYDIR	Windows System directory	C:\WINDOWS\SYSTEM\
ISWINDISK	Disk containing the Windows directory	C:
ISWINDIR	Windows directory	C:\WINDOWS\
ISSRCDISK	Disk where the installation files are stored	D:
ISSRCDIR	Directory where the installation files are stored	D:\INSTALLER\VirusScan30\
ISSUPPORTDIR	Directory where the install puts its temporary files	C:\TEMP\ISTMP11.DIR\
ISSVTARGET	Directory where the product will be installed	C:\Program Files\McAfee\VirusScan\
ISCOMMONFILES	Where Windows common files are located	C:\Program Files\Common Files\
ISFOLDER_DESKTOP	Where the Desktop folder is located	C:\WINDOWS\Desktop\
ISFOLDER_STARTMENU	Where the Start Menu folder is located	C:\WINDOWS\Start Menu\
ISFOLDER_STARTUP	Where the StartUp folder is located	C:\WINDOWS\Start Menu\Programs\StartUp\
ISPROGRAMFILES	Where the Program File folder is located	C:\Program Files\

Examples using variables

- To copy the licensing agreement from %ISSRCDIR% to %ISSUPPORTDIR% (using our example data), translate “Copy D:\INSTALLER\VirusScan30\License.txt C:\TEMP_ISTMP11.DIR”:

```
CopyFile( " ",
          "LICENSE.TXT",
          "%ISSRCDIR%",
          " ",
          "%ISSUPPORTDIR%",
          0 );
```

- To extract the *.DAT files from the file DATS.ZIP stored in %ISSRCDIR% to %ISSVTARGET% (using our example data), translate “Copy D:\INSTALLER\VirusScan30\dats.zip extract *.DAT to C:\Program Files\McAfee\VirusScan”:

```
CopyFile( "%ISSRCDIR%DATS.ZIP",
          "*.DAT",
          " ",
          " ",
          "%ISSVTARGET%",
          0 );
```

A trailing backslash (\) is already embedded in the variables, so you don’t need to add one. If you do add an additional backslash, the compiler will interpret this string with two backslash characters, and the command will fail.

Pay attention to the logic of your scripts and the order of their actions. For example, notice that using the “ISSVTARGET” variable before the “PreGetDest()” subject header causes SETUP.EXE to fail because the destination of the installation is not specified.

Setup Command-line Options


SISSRC and SISABS

Use this feature to specify which scripts you want the install to run once its launched. For example, you might want different departments within your company to access different .SIS files during login. Maybe you want to provide a .SIS file that would only be used if some shell application is launching the install. Or, perhaps you are using a NetWare login script that has two different paths, each with its own .SIS file. You can specify additional .SIS files using these two switches.

 *For details on creating valid .SIS filenames, see “ISeamless Quick Start” on page 11, or “Using the Compiler” on page 47.*

SISSRC assumes the .SIS file is in the install source directory (where SETUP.EXE is located). SISABS expects an absolute path to the file.

You can use these parameters as many times as you want in the same script. You can even mix and match them on the same command line.

 *Make sure these are the first parameters on the command line. If you use any additional flags for the setup (-s, -SMS, etc.), make sure they follow any SISABS or SISSRC parameters.*

Sample commands

- Run setup in normal mode using the EXTRA1.SIS file located in C:\TEST:

```
setup SISABS"C:\TEST\EXTRA1.SIS"
```

- Run setup in silent mode using the EXTRA1.SIS file located in C:\TEST:

```
setup SISABS"C:\TEST\EXTRA1.SIS" -s
```

- Run setup in normal mode using the "COPY THE EXTRA FILES FOR NETWARE.SIS" file located in the source directory (i.e. %ISSRCDIR%):

```
setup SISSRC"COPY THE EXTRA FILES FOR NETWARE.SIS"
```

- Run setup in record mode using the two .SIS files mentioned on the command line:

```
setup SISABS"C:\TEST\EXTRA INFO.SIS" SISSRC"COPY  
SUPPORT FILES.SIS" -r
```

5

Examining Three Examples

Before reading “[Defining Actions](#)” on [page 24](#), which details ISeamless actions and their variables, study the original three examples on [page 5](#) in terms of actions and phases using the terms you’ve learned so far. When deciding how you want to deploy your anti-virus or network security software, you will need to do the same thing—turn your ideas into a mathematical word problem. Studying these examples is an integral step, and will greatly reduce trial and error in your scripting as you master ISeamless.

1. Read the example problems in terms of scripted actions and phases.

Example 1: Before the licensing agreement appears, copy a custom licensing agreement for the install to display. After the files are copied, copy (overwrite) the original licensing agreement on the hard drive with the custom one.

Example 2: After the files are copied, copy (overwrite) the DEFAULT.VSC and DEFAULT.VSH on the hard drive with a custom set of .VSC/.VSH files. After the product has been configured, set a registry key.


Example 3: After the files are copied, copy (overwrite) the .DAT files on the hard drive with the custom .DATs.

2. After you convert your idea into a word algorithm, translate the algorithm into ISeamless scripting codes.

Example 1:

```
PreLicense
{
    CopyFile( " ",
              "LICENSE.TXT",
              "%ISSRCDIR%",
              " ",
              "%ISSUPPORTDIR%",
              0 );
}

PostCopy
{
    CopyFile( " ",
              "LICENSE.TXT",
              "%ISSRCDIR%",
              " ",
              "%ISSVTARGET%",
              0 );
}
```

 Although ISeamless uses C code as the syntax for its commands, the compiler only supports about 5 percent of all C code constructs and functionality.

All of the following ISeamless scripting commands work under Windows 95 and Windows NT. However, scripts containing the newer commands DeleteRegKey, DeleteRegValue, or FileAttribute do not work under Windows 3.x. Many of the commands (FileAttribute, EnterDisk, DeleteRegValue, DeleteRegKey, AddGroupIcon, DeleteGroupIcon, and DeleteGroup) only work with the latest ISeamless compiler, version 1.2.0.

CopyFile

Use this command to copy or extract files from a .ZIP archive.

Sample syntax:

```
CopyFile( STRING SourceZipFile,  
          STRING SourceFileName,  
          STRING SourceDirectory,  
          STRING TargetFile,  
          STRING TargetDirectory,  
          NUMBER Recursion );
```

- **SourceZipFile [STRING]**—specifies the name of the .ZIP file from which you want to extract the files. If you specify "", the CopyFile command will not use a .ZIP file and will take the SourceDirectory as the source location. Include the directory of the .ZIP file in this parameter.
- **SourceFileName [STRING]**—specifies the name of the file you want to copy. If you are using a .ZIP file, you can use the following types of file names: Filename.EXT, LongFileName.Extension, *.Extension, Filename.*, or *.* If you are not, you can use any file specification.
- **SourceDirectory [STRING]**—specifies the location of the file to be copied. If you are using a .ZIP file, then this will be the source directory contained in the .ZIP file you are extracting from. If you are not, it's the location of the SourceFileName.
- **TargetFile [STRING]**—is a currently unavailable command. You must enter an empty string ("").
- **TargetDirectory [STRING]**—specifies the location of the copied or extracted files.
- **Recursion [INTEGER]**—performs the same copy operation on every sub-directory contained in the specified source directory. To enable this task, type "1." Otherwise, type "0" to recreate the same sub-directory tree of your SourceDirectory in your TargetDirectory.

Sample commands

- Copy the AUTOEXEC.BAT from C:\ to C:\WINDOWS\TEMP:

```
CopyFile( " ",  
          "AUTOEXEC.BAT" ,  
          "C:\\",  
          " ",  
          "C:\\WINDOWS\\TEMP" ,  
          0 );
```

- Extract LICENSE.TXT from LICENSE.ZIP stored in C:\ZIP, and copy to C:\PROGRA~1\MCAFEЕ:

```
CopyFile( "C:\\ZIP\\LICENSE.ZIP" ,  
          "LICENSE.TXT" ,  
          "C:\\",  
          " ",  
          "C:\\PROGRA~1\\MCAFEЕ" ,  
          0 );
```

- Extract the *.DAT files from the file DATS.ZIP stored in C:\ZIP and copy to C:\PROGRA~1\MCAFEЕ. Note the use of a directory structure inside the .ZIP file:

```
CopyFile( "C:\\ZIP\\DATS.ZIP" ,  
          "*.DAT" ,  
          "FEB\\",  
          " ",  
          "C:\\PROGRA~1\\MCAFEЕ" ,  
          0 );
```

DeleteFile

Use this command to delete a file or a set of files.

Sample syntax:

```
DeleteFile( STRING SourceFile,  
            STRING SourceDestination,  
            INTEGER ExtraFlag );
```

- **SourceFile [STRING]**—specifies either an individual filename or a set of files (wildcards).
- **SourceDestination [STRING]**—specifies the location where the files are stored.
- **ExtraFlag [INTEGER]**—is a currently unavailable command. Enter “0”.

Sample commands

- Delete the AUTOEXEC.BAT located in C:\:

```
DeleteFile( "AUTOEXEC.BAT",  
            "C:\\",  
            0 );
```

- Delete all ~*.tmp files from the C:\TEMP directory:

```
DeleteFile( "~*.tmp",  
            "C:\\TEMP",  
            0 );
```

- Delete LICENSE.TXT from the %ISSVTARGET% directory (using the example data); delete C:\Program Files\McAfee\VirusScan\LICENSE.TXT:

```
DeleteFile( "LICENSE.TXT",  
            "%ISSVTARGET%",  
            0 );
```

RunCustomProgram


Use this command to run selected programs while the install is running.

Sample syntax:

```
RunCustomProgram( STRING ProgramPathAndFile,  
                  STRING Arguments,  
                  NUMBER Wait );
```

- **ProgramPathAndFile [STRING]**—specifies the directory path and filename for the program you want to run.
- **Arguments [STRING]**—specifies the arguments you pass to the program. If there are no arguments in your script, type an empty string ("").
- **Wait [NUMBER]**—specifies whether or not the install waits after running your selected application. To make the install wait for your program's completion before resuming, type "1". To make the install launch your program and continue the install without waiting, type "0".

If you type "0" (no wait), be sure the program will not interfere with the install. If you have problems with the install, try typing "1" (wait), or do not launch any external programs.

 *The effect of the wait parameter can be tricky: For example, if you type "1" for the install to wait after launching NOTEPAD.EXE, the install will halt until you close NOTEPAD.EXE. Also, your program's setup is unaware of any child processes launched by the parent process specified by the RunCustomProgram command. So, if you specify a wait parameter of "1", the install will only wait for the completion of the parent process before resuming.*


SetRegValue

Use this command to modify registry values.

Sample syntax:

```
SetRegValue( STRING HKEY_Source,
             STRING Key,
             STRING Name,
             STRING Type,
             STRING Value,
             NUMBER ExtraParam );
```

- HKEY_Source [STRING]—represents the starting point in the tree. The source must be one of the following: "HKEY_LOCAL_MACHINE", "HKEY_CLASSES_ROOT", "HKEY_CURRENT_USER", or "HKEY_USERS".
- Key [STRING]—indicates the name of the registry key.
- Name [STRING] —indicates the name associated with the value.
- Type [STRING]—indicates the three value types: STRING, binary, or DWORD. Type "STRING" if the value you are going to enter in "Value [STRING]" is an alphanumeric entry. Type "NUMBER" if the value you are going to enter in "Value [STRING]" is a DWORD, i.e. 0x00000000 (0).

 *ISeamless does not support binary entries at this time. For more information about the different type of registry entries, consult the documentation available at the Microsoft website (<http://www.microsoft.com>).*

- Value [STRING]—indicates the value of either your "STRING" or "NUMBER" entry. A STRING may be alphanumeric, or may contain only letters, only numbers, or both. If a STRING contains only numbers, you must put quotation marks (" ") around the value. If you select a STRING, and the STRING is a set of numbers not end-capped by quotation marks (" "), the compiler tries to interpret the parameter as a real number instead of a string, and fails.
- ExtraParam [NUMBER]—is a currently unavailable command. Enter "0".

Sample commands

- Set the bScanAllFile DWORD Value to “1”. It’s located under HKEY_LOCAL_MACHINE\SOFTWARE\MCAFFEE\SCREENSCAN:

```
SetRegValue( "HKEY_LOCAL_MACHINE",  
             "\\SOFTWARE\\MCAFFEE\\SCREENSCAN",  
             "bScanAllFiles",  
             "NUMBER",  
             "1",  
             0 );
```

- Set the szLogFileName String Value to C:\LOG.TXT. It’s located under HKEY_LOCAL_MACHINE\SOFTWARE\MCAFFEE\SCREENSCAN:

```
SetRegValue( "HKEY_LOCAL_MACHINE",  
             "\\SOFTWARE\\MCAFFEE\\SCREENSCAN",  
             "szLogFileName",  
             "STRING",  
             "C:\\LOG.TXT",  
             0 );
```

SetIniValue

This command is very similar to SetRegValue. Use this function to modify .INI file settings made by the install.

```
GetSetIniValue( STRING File,  
                STRING Section,  
                STRING Key,  
                STRING Value,  
                NUMBER ExtraParam );
```

- File [STRING]—indicates the .INI filename and directory path for the entry you want to change.
- Section [STRING]—indicates the section in which you want to modify a value.
- Key [STRING]—indicates the name associated with that value.
- Value [STRING]—indicates the value itself. Even if it's a number, the value must be in quotation marks (" ").
- ExtraParam [INTEGER]—is a currently unavailable command. Enter "0".

Sample commands


- Change the ScrnSave.exe entry in the boot section of SYSTEM.INI to %ISWINSYSDIR%FLYING~2.SCR:

```
SetIniValue( "%ISWINDIR%SYSTEM.INI",  
             "boot",  
             "ScrnSave.EXE",  
             "%ISWINSYSDIR%FLYING~2.SCR",  
             0 );
```

- Change the szCustomMessage entry in the AlertOptions section of DEFAULT.VSC located in the target directory to “Please call...”:

```
SetIniValue( "%ISSVTARGET%DEFAULT.VSC",  
             "AlertOptions",  
             "szCustomMessage",  
             "Please call helpdesk at x1234",  
             0 );
```


FileAttribute


 *This function works only with ISeamless v1.2.0 and later. It does not work with Windows 3.x.*

Use this function to modify the attributes of files, e.g. to make some files read-only after the install.

Sample syntax:

```
FileAttribute( STRING SourceDirectory,  
              STRING FileName,  
              STRING Attributes,  
              NUMBER ExtraParam );
```

- SourceDirectory [STRING]—indicates the directory where the file you want to change is located.
- FileName [STRING]—indicates the name of the file. This can either be a valid filename or “*.*” (wildcard).
- Attributes [STRING]—indicates the attributes to add or remove. You can change read-only (R), hidden (H), system (S) or archive (A) attributes. To add attributes, type a plus “+” followed by the corresponding attribute letter. To remove attributes, type a “-” followed by the corresponding attribute letter. You can set more than one attribute at a time by appending them into a single string; order does not matter.

 *If you do not mention an attribute in your call, the attribute remains the same. For example, if you don't include a reference to the H (hidden) attribute, the H attribute remains unchanged.*

- ExtraParam [NUMBER]—specifies whether to recursively modify every file and directory contained in SourceDirectory. If you entered “*.*” as the FileName, type “1” here to enable this function.

Sample commands

- Turn on the read-only attribute of the file AUTOEXEC.BAT located on the root of the C drive:

```
FileAttribute( "C:\\",  
              "AUTOEXEC.BAT",  
              "+R",  
              0 );
```

- Turn off any attributes of the file NOTEPAD.EXE located in the Windows directory:

```
FileAttribute( "%ISWINDIR%",  
              "NOTEPAD.EXE",  
              "-R-S-H-A",  
              0 );
```


- Turn off any attributes of the file NOTEPAD.EXE located in the Windows directory. Rearrange the order of the attributes (you'll get the same result as above):

```
FileAttribute( "%ISWINDIR%",  
              "NOTEPAD.EXE",  
              "-H-A-R-S",  
              0 );
```

- Set all files in the installed directory to R (Read-only), and remove the A (Archive) attribute. Notice the use of the ExtraParam and the fact you can mix addition and removal of attributes:

```
FileAttribute( "%ISSVTARGET%",  
              "*. *",  
              "+R-A",  
              0 );
```

EnterDisk

 *This function works only with ISeamless v1.2.0 and later. It does not work with silent installs.*

Use this command to ask the user to insert a floppy disk, e.g. when you want the user to insert a disk at the end of a floppy install to copy additional files from.

Sample syntax:

```
EnterDisk( STRING Message,  
          STRING FileToVerify,  
          NUMBER ExtraParam );
```


- **Message [STRING]**—represents the message you want displayed in the window that appears for the user during the install. Write a clear message telling the user which disk to insert.
- **FileToVerify [STRING]**—indicates the name of the file included on the installation disk that will authenticate the disk.
- **ExtraParam [INTEGER]**—is a currently unavailable command. Enter “0”.

Sample command

Ask for a disk labeled “Additional” containing the file DATFILES.ZIP:

```
EnterDisk( "Please enter the disk labeled Additional",  
          "DATFILES.ZIP" );
```

DeleteRegValue

 *This function works only with ISeamless v1.2.0 and later. It does not work with Windows 3.x.*


Use this command to delete registry values.

Sample syntax:

```
DeleteRegValue( STRING HKEY_Source ,  
                STRING Key ,  
                STRING Name ,  
                NUMBER ExtraParam );
```

- HKEY_Source [STRING]—represents the starting point in the tree. The source must be one of the following: "HKEY_LOCAL_MACHINE", "HKEY_CLASSES_ROOT", "HKEY_CURRENT_USER", or "HKEY_USERS".
- Key [STRING]—indicates the name of the key.
- Name [STRING]—indicates the name associated with the value.
- ExtraParam [INTEGER]—is a currently unavailable command. Enter "0".

Sample commands

 *These sample commands are shown only as hypothetical examples of a DeleteRegValue calls. These commands could disable a product that uses them, so make sure you delete any necessary values.*


- Delete the bScanAllFile Value located under HKEY_LOCAL_MACHINE\SOFTWARE\MCAFFEE\SCREENSCAN:

```
DeleteRegValue( "HKEY_LOCAL_MACHINE" ,  
                "\\SOFTWARE\\MCAFFEE\\SCREENSCAN" ,  
                "bScanAllFiles" ,  
                0 );
```

- Delete the szLogFileName value located under
HKEY_LOCAL_MACHINE\SOFTWARE\MCAFFEE\SCREENSCAN:

```
DeleteRegValue( "HKEY_LOCAL_MACHINE",  
                "\\SOFTWARE\\MCAFFEE\\SCREENSCAN",  
                "szLogFileName",  
                0 );
```

DeleteRegKey

 *This function works only with ISeamless v1.2.0 and later. It does not work with Windows 3.x.*


Use this command to delete registry keys.

Sample syntax:

```
DeleteRegKey( STRING HKEY_Source ,  
              STRING Key ,  
              NUMBER ExtraParam );
```

- HKEY_Source [STRING]—represents the starting point in the tree. The source must be one of the following: "HKEY_LOCAL_MACHINE", "HKEY_CLASSES_ROOT", "HKEY_CURRENT_USER", or "HKEY_USERS".
- Key [STRING]—indicates the name of the key.
- ExtraParam [INTEGER]—is a currently unavailable command. Enter "0".

Sample command

 *This sample command is shown only as an example of a DeleteRegValue call. These commands could disable a product that uses them, so make sure you delete any necessary values.*

Delete the HKEY_LOCAL_MACHINE\SOFTWARE\MCAFFEE\SCREENSCAN key:

```
DeleteRegKey( "HKEY_LOCAL_MACHINE" ,  
              "\\SOFTWARE\\MCAFFEE\\SCREENSCAN" ,  
              0 );
```

AddGroupIcon

 *This function works only with ISeamless v1.2.0 and later.*

Use this command to add icons to program groups if the install didn't create them for you.

Sample syntax:

```
AddGroupIcon( STRING ProgramFolder,
               STRING ItemName,
               STRING Program,
               STRING Parameters,
               STRING WorkingDirectory,
               NUMBER ExtraParam );
```

- **ProgramFolder [STRING]**—specifies the name of the program group to add an icon to. If you are running Windows 95 or Windows NT 4.0, use %ISSVFOLDER%, %ISFOLDER_DESKTOP%, %ISFOLDER_STARTMENU%, or %ISFOLDER_STARTUP% to specify the program group's location.
- **ItemName [STRING]**—indicates the name of the icon.
- **Program [STRING]**—indicates the full path to the program's executable file.
- **Parameters [STRING]**—indicates the parameters you want passed to the program's executable file.
- **WorkingDirectory [STRING]**—indicates the location to run the program from. To use the default directory, enter "".
- **ExtraParam [INTEGER]**—is a currently unavailable command. Enter "0".

Sample command

Add a shortcut to NOTEPAD.EXE by editing AUTOEXEC.BAT in the StartUp group:

```
AddGroupIcon( "%ISFOLDER_STARTUP%",  
               "Edit AUTOEXEC.BAT",  
               "%ISWINDIR%NOTEPAD.EXE",  
               "C:\\AUTOEXEC.BAT",  
               "",  
               0 );
```


DeleteGroupIcon

 *This function works only with ISeamless v1.2.0 and later.*

Use this command to delete icons created by the install from program groups.

Sample syntax:

```
DeleteGroupIcon( STRING ProgramFolder,  
                 STRING ItemName,  
                 NUMBER ExtraParam );
```

- ProgramFolder [STRING]—specifies the name of the program group you want to modify. If you are running Windows 95 or Windows NT 4.0, use %ISSVFOLDER%, %ISFOLDER_DESKTOP%, %ISFOLDER_STARTMENU% or %ISFOLDER_STARTUP% to specify the program group's location. With Windows 3.x, use %ISSVFOLDER%.
- ItemName [STRING]—indicates the name of the icon.
- ExtraParam [INTEGER]—is a currently unavailable command. Enter "0".

Sample command

Delete the shortcut to "Read Me 1st" located in the software's program group:

```
DeleteGroupIcon( "%ISSVFOLDER%",  
                 "Read Me 1st",  
                 0 );
```

DeleteGroup

 *This function works only with ISeamless v1.2.0 and later.*

Use this command to delete program groups that the install created.

Sample syntax:

```
DeleteGroup( STRING ProgramFolder,  
             NUMBER ExtraParam );
```

- ProgramFolder [STRING]—specifies the name of the program folder you want remove. If you are running Windows 95 or Windows NT 4.0, use %ISSVFOLDER%, %ISFOLDER_DESKTOP%, %ISFOLDER_STARTMENU%, or %ISFOLDER_STARTUP% to specify the program group's location. With Windows 3.x, use %ISSVFOLDER%.
- ExtraParam [INTEGER]—is a currently unavailable command. Enter "0".

Sample command

Delete the software's program group:

```
DeleteGroup( "%ISSVFOLDER%",  
             0 );
```

VariableValue

Use this command to configure settings for the install.

Sample syntax:

```
VariableValue( STRING Variable,  
              STRING Value );
```

Each of the three supported variables (NCCONFIGDEBUG, ISSILENT_FINISHDELAY, and ISSILENT_ABNORMAL) has a predefined set of associated values.

NCCONFIGDEBUG

Use this variable to enable the script debugger. By default, the debugger is off. You can change the state of the debugger more than one time in the script.

 *The scripting debugger does not run at the same time as the silent install.*

Value	Meaning
TRUE	Enables scripting debugging
FALSE	Disables scripting debugging

Sample commands

Turn the debugger on:


```
VariableValue( "NCCONFIGDEBUG",  
              "TRUE" );
```

Turn the debugger off:

```
VariableValue( "NCCONFIGDEBUG",  
              "FALSE" );
```

NNEEDREBOOT

Use this variable to determine whether to display a window with a reboot option at the end of the install, or a window that simply says the install is complete. This is very useful if you are running the install as part of a suite of other installations and you want to handle the reboot process yourself.

 *You must only use this command in the **PreFinish** section of the installation.*

Value	Meaning
TRUE	Displays the window that lets the user decide whether or not to reboot
FALSE	Displays the window that only says the installation is complete

Sample commands

- Display the window containing the option to reboot at the end:

```
VariableValue( "NNEEDREBOOT",  
              "TRUE" );
```

- Display the window that only says the installation is complete:

```
VariableValue( "NNEEDREBOOT",  
              "FALSE" );
```

ISSILENT_FINISHDELAY

Use this variable to customize the amount of time the Silent Install Notification window remains open if a silent install is incomplete. By default, the window stays open for 30 seconds. If a silent install succeeds, the window automatically disappears.

Here are the possible values for the silent install notification variable:

Value	Meaning
-1	Leaves the window open until the user closes the program
0	Closes the window immediately
Any positive value (15, 30, 100, etc.)	Specifies the number of seconds to leave the window open

Sample commands

- Force the program window to stay open until the user closes it:

```
VariableValue( "ISSILENT_FINISHDELAY",  
              "-1" );
```

- Force the program window to automatically close when the install fails:

```
VariableValue( "ISSILENT_FINISHDELAY",  
              "0" );
```

- Force the program window to stay open for about 60 seconds:

```
VariableValue( "ISSILENT_FINISHDELAY",  
              "60" );
```

ISSILENT_ABNORMAL

Use this variable to customize the message that appears if the silent install fails. The default message is “Silent installation failed,” which appears in the silent install animated window and icon.

Sample commands

- Display the message “Please call extension 1234 to report the failed install” if the install fails:

```
VariableValue( "ISSILENT_ABNORMAL",  
  "Please call extension 1234 to report the failed install" );
```

- Display no message if the install fails:

```
VariableValue( "ISSILENT_ABNORMAL",  
  "" );
```

Use the ISeamless compiler to translate your custom ISeamless install script. The compiler consists of three files, which must be in the same directory:

- ISEAMCMP.EXE
- ISEAM.LLR
- ISEAM.DFA

This compiler is slightly different than what you would normally use to compile computer code. A compiler changes programming code, a language that humans can understand and write, into machine language, which is a combination of 0s and 1s. After writing your code, you translate it using a compiler and a linker into an executable program usable by your computer.

The compiler for ISeamless works similarly. After you've written your .ISC file using ISeamless scripting codes, you translate it using the compiler into an .SIS file, which is usable by your software's SETUP.EXE.


Follow these steps to compile a script file:

- | Step | Action |
|-------------|---|
| 1. | From your Windows 95 or Windows NT Start menu, point to Programs, then click MS-DOS Prompt. |
| 2. | Change to the directory that contains ISEAMCMP.EXE. |

3. Type:


```
ISEAMCMP FILENAME.ISC NEWNAME.SIS
```

Response: A new file called NEWNAME.SIS appears in the current directory.

 *If compilation fails, error messages appear. For a list of possible error messages and their definitions, see [Appendix B, "Compilation Error Codes."](#) For brief descriptions of the commands and variables that can be used in the source file, see [Chapter 4, "Understanding ISeamless."](#) For detailed definitions of ISeamless commands and variables, see [Chapter 6, "Defining Actions."](#)*

- FILENAME.ISC is the name of the script file you want to compile. The compiler creates a new file with the default name ADMIN.SIS in the current directory.
- NEWNAME.SIS is an optional command that lets you give the compiled file a different name than the default name ADMIN.SIS. In place of NEWNAME, enter an 8-character name (or a long filename bound with quotes).


4. Rename the file with a valid .SIS filename. Valid file names for automatic installs are ADMIN.SIS, OEM.SIS, and AUTO*.SIS (any file starting with AUTO and having .SIS as its extension).

 *If you name the .SIS file with an invalid filename for automatic installs, you must use SISSRC or SISABS as the first parameter on your setup command line.*

For CD installs, you can choose any filename.

OR

For floppy installs, you must name the file ADMIN.SIS and OEM.SIS.

 *Different departments within your company may have different protection needs. If you use this option to deploy your software with different configurations, giving each .SIS file a unique name will help you to manage them.*

5. Copy or move the newly compiled .SIS file to the source directory of your install (%ISSRCDIR%), where your software's SETUP.EXE and other installation files are located.
6. Copy any additional files needed by your script.
7. Run SETUP.EXE, in silent or non-silent mode.

Here are sample scripts based on the three examples. Mix and match parts of these scripts to make your own.

Using a Custom Licensing Agreement

```
PreInitial()  
{  
    // VariableValue(  
    // Uncomment if you want the debugger on.  
    // "NCCONFIGDEBUG",  
    // "TRUE" );  
}  
PreLicense()  
{  
    /* The next section extracts the licensing agreement  
    from a .ZIP file located in the same directory as the  
    install. This way, the custom agreement is displayed  
    during setup instead of the standard one. */  
    CopyFile( "%ISSRCDIR%License.zip",  
              "LICENSE.TXT",  
              "  
              "  
              "%ISSUPPORTDIR%",  
              0 );  
}
```

```

PostLicense()
{
    // This is an optional section, which just shows you
    // how to erase a file.
    DeleteFile( "LICENSE.TXT",
               "%ISSUPPORTDIR%",
               0 );
}
PostCopyFile()
{
    // You also need to overwrite the agreement that gets
    // copied onto the disk.
    CopyFile( "%ISSRCDIR%License.zip",
             "LICENSE.TXT",
             "",
             "",
             "%ISSVTARGET%",
             0 );
}

```

Using Custom .VSC and .VSH Files

```

// This simple script extracts all the default files
// from the DEFAULT.ZIP file located in the source
// directory.
PostCopyFile()
{
    CopyFile( "%ISSRCDIR%Default.zip",
             "DEFAULT.*",
             "",
             "",
             "%ISSVTARGET%",
             0 );
}

```

Using Newer .DAT Files

```
// This script puts different .DAT files in the target
// directory.
PostCopyFile()
{
    CopyFile( "%ISSRCDIR%Dats.zip",
              "*.DAT",
              " ",
              " ",
              "%ISSVTARGET%",
              0 );
}
```

Compilation Error Codes

If compilation fails, error messages appear. Most error messages contain a line number preceding the error message. This line number represents the line where the error was detected; however, the error in syntax may not be precisely on that line, but on a line preceding it.

Try to resolve errors in the order they are found. Often, one error can generate other errors, and resolving the first error may resolve them all.

C1001 - The type of argument is invalid: <KEYWORD>

The compiler finds a string where an integer is expected, or an integer where a string is expected.

C1002 - Only new section allowed here: <KEYWORD>

The compiler finds the <KEYWORD> where a new section name is expected. Check the spelling of the <KEYWORD>, and make sure a close brace “}” exists.

C1003 - Only new command allowed here: <KEYWORD>

The compiler finds the <KEYWORD> where a new command name is expected. Check the spelling of the <KEYWORD>.

C1004 - Only end of command allowed here: <KEYWORD>

The compiler finds the <KEYWORD> where “)” is expected. Check the number of parameters and the syntax.

C1005 - Only beginning of command or end of section allowed here: <KEYWORD>

The compiler finds the <KEYWORD> where a new command or end of section “)” is expected. Check the spelling of the <KEYWORD>.

C1006 - Too many arguments for this command: <KEYWORD>

The compiler finds the <KEYWORD> where “)” is expected. Check the number of parameters and the syntax.

C1007 - Too few arguments for this command: <KEYWORD>

The compiler finds the <KEYWORD> where more parameters are expected. Check the number of parameters and the syntax.

C1008 - Invalid command used: <KEYWORD>

The compiler finds the <KEYWORD> where a new command name is expected. Check the spelling of the <KEYWORD>.

C1009 - Invalid section used: <KEYWORD>

The compiler finds the <KEYWORD> where a new section name is expected. Check the spelling of the <KEYWORD>.

C2000 - Syntax error(s) detected in compilation of <KEYWORD>

Syntax errors occur when you don't adhere to the structure of the scripting language. You can avoid most common syntax errors by ensuring that you do the following:

- Use a semicolon “;” at the end of a statement.
- If you use an open brace “{”, use a close brace “}”.
- Use a comma “,” between parameters.
- If you use an open parenthesis “(”, use a close parenthesis “)”.
- If you use an open comment “/*”, use a close comment “*/”.

C2001 - Major errors for %s detected. Please check the environment and the existence of the file.

This error may have little to do with the content and syntax of the script; it's possible that your file's path is incorrect. Here are some possible reasons for this error message:

- The file you're trying to compile does not exist.
- The file you're trying to compile is locked.
- The ISEAM.LLR file is missing or not in the same directory as the compiler.
- The ISEAM.DFA file is missing or not in the same directory as the compiler.
- Your computer lacks enough memory to run the compiler. Close down some applications and try again.

C2002 - There were errors during compilation. Please correct them and try again.

Errors were detected, but they are not syntax errors.

B

Bulletin Board System
(BBS) 7

C

Command-line options
 SISSRC and SISABS
 20

Commands

 actions 17

 AddGroupIcon 39

 CopyFile 25

 DeleteFile 27

 DeleteGroup 42

 DeleteGroupIcon
 41

 DeleteRegKey 38

 DeleteRegValue
 36

 EnterDisk 35

 FileAttribute 33

 RunCustomProgram
 28

 SetIniValue 31

 SetRegValue 29

 VariableValue 43,
 44, 45, 46

 directory locations 18

 timing 16

Compiled file

 invalid filenames

 SISSRC and
 SISABS
 parameters 12,
 20, 48

 valid filenames

 ADMIN.SIS 12, 48

 AUTO*.SIS 12, 48

 OEM.SIS 12, 48

Compiler

 C code support 13

 error codes 53

 required files

 ISEAM.DFA 47

 ISEAM.LLR 47

 ISEAMCMP.EXE
 47

 syntax 16

 syntax requirements
 15

 system requirements
 11

 text editor 13

 using 47

Custom .VSC and .VSH
 files 5, 22, 51

Custom licensing
 agreement 5, 22, 50

Customer Care

 contacting 7

D

Data files

 distributing newer 5,
 22, 52

E

Error codes

 definitions 53

I

ISeamless

 compiler system
 requirements 11

 definition of 5

 general syntax
 information 15

 introducing 5

 quick start overview 11

 script system
 requirements 11

 syntax 16

 updating installation
 .DAT files 13

 using 11

Installation

- custom .VSC and .VSH files [5](#), [22](#), [51](#)
- custom licensing agreement [5](#), [22](#), [50](#)
- newer .DAT files [5](#), [22](#), [52](#)

N

Network Associates

- contacting
 - BBS [7](#)
 - Customer Care [7](#)
 - outside the United States [9](#)
 - via America Online [7](#)
 - via CompuServe [7](#)
 - within the United States [8](#)
- training [8](#)

O

Options

- command-line [20](#)

S

Script

- definition [6](#)

Scripts

- compiling [47](#)
- samples [50](#)
- syntax [16](#)
- system requirements [11](#)
- using a custom licensing agreement [5](#), [22](#), [50](#)
- using a newer set of .DAT files [5](#), [13](#), [22](#), [52](#)
- using custom .VSC and .VSH files [5](#), [22](#), [51](#)
- writing [13](#)

Setup command-line

- options [12](#), [20](#), [48](#)

Syntax

- action commands [17](#)
- directory location commands [18](#)
- general [15](#)
- scripting [16](#)
- timing commands [16](#)

T

Technical support

- Bulletin Board System (BBS) [7](#)
- e-mail address [7](#)
- information needed from user [8](#)
- online [7](#)

Training

- scheduling [8](#)

Training for Network

- Associates products [8](#)

V

VariableValue command

- ISSILENT_ABNORMAL [46](#)
- ISSILENT_FINISHDEL AY [45](#)
- NCCONFIGDEBUG [43](#)
- NNEEDREBOOT [44](#)