

Telelogic Logiscope

Microsoft Visual C++.NET Integration

Version 6.5

Before using this information, be sure to read the general information under “Notices” section, on page 9.

This edition applies to **VERSION 6.5, TELELOGIC LOGISCOPE** (product number 5724V81) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1985, 2008**

US Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

About this manual

Audience

This reference manual is intended for **Telelogic® Logiscope™** users such as software developers, project managers or quality engineers who want to use the Logiscope toolset within the **Microsoft Visual C++ .NET** environment.

Overview

This manual is a complement to the **Telelogic Logiscope** documentation.

Section 1 explains what is the content of the integration.

Section 2 explains how to install the integration.

Section 3 tours the available functionalities of the integration.

Conventions

The following typographical conventions are used in this manual:

<i>italics</i>	names of textual elements (filename), notes, documentation titles.
typewriter	screen and file examples.

Contacting IBM Rational Software Support

Support and information for Telelogic products is currently being transitioned from the Telelogic Support site to the IBM Rational Software Support site. During this transition phase, your product support location depends on your customer history.

Product support

- If you are a heritage customer, meaning you were a Telelogic customer prior to November 1, 2008, please visit the [Logiscope Support Web site](#).

Telelogic customers will be redirected automatically to the IBM Rational Software Support site after the product information has been migrated.

- If you are a new Rational customer, meaning you did not have Telelogic-licensed products prior to November 1, 2008, please visit the [IBM Rational Software Support site](#).

Before you contact Support, gather the background information that you will need to describe your problem. When describing a problem to an IBM software support specialist, be as specific as possible and include all relevant background information so that the specialist can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, or messages that are related to the problem?
- Can you reproduce the problem? If so, what steps do you take to reproduce it?
- Is there a workaround for the problem? If so, be prepared to describe the workaround.

Other information

For Rational software product news, events, and other information, visit the [IBM Rational Software Web site](#).

Bibliography

[TCL94] JOHN K. OUSTERHOUT

Tcl and the Tk Toolkit - Addison-Wesley Professional Computing Series
1994 ISBN 0-201-63337-X

[TCL03] BRENT WELCH, KEN JONES, JEFFREY HOBBS

Practical Programming in Tcl and Tk (4th Edition) – Prentice Hall
2003 ISBN 0-130-38560-3

Table of Contents

1. Overview.....	1
2. Installation.....	2
3. Using Logiscope from Visual C++.....	4
3.1. Building Logiscope QualityChecker Results.....	4
3.2. Building Logiscope RuleChecker Results.....	4
3.3. Building Logiscope Instrumented binary.....	5
3.4. Launching Logiscope Studio.....	6
3.5. Customizing the integration.....	6

1. Overview

The integration allows access Logiscope functionalities from Microsoft Visual C++ .NET graphical environment.

This integration is available for C and C++ Visual C++ projects.

A Visual C++ project shall be considered as a C project if it contains only C source files. A Visual C++ project containing at least one C++ source file shall be considered as a C++ project.

Once the integration is installed it is possible to:

- Build and visualize *QualityChecker* results (e.g. source code metrics, components complexity rating) from the Microsoft Visual C++.NET graphical environment.
- Check rules and visualize violations from the Microsoft Visual C++.NET graphical environment.
- Build instrumented binaries (for coverage analysis) from the Microsoft Visual C++.NET graphical environment.
- Launch Logiscope Studio for a more comprehensive use of Telelogic Logiscope Toolset.
- Edit the configuration file for default settings (for instance the set of rule to check).

The integration has been developed and tested with :

- Microsoft Visual C++ .NET 2003 (Microsoft Development Environment 2003 Version 7.1.3088).
- Microsoft Visual C++ .NET 2005 (Microsoft Development Environment 2005 Version 8.0.50727.42).

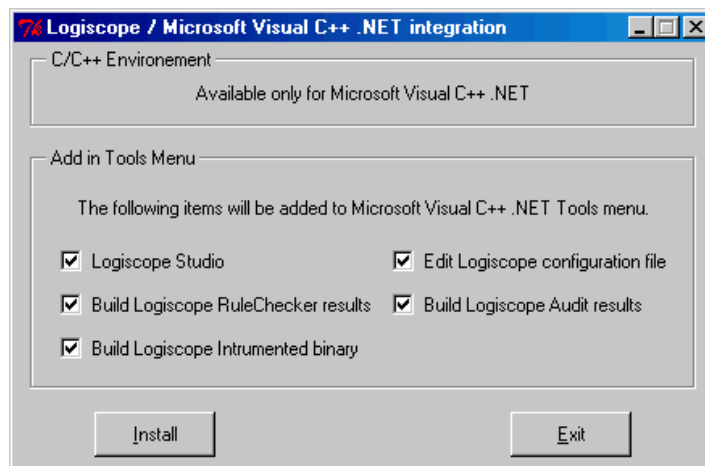
Known Limitations:

- The Logiscope integration does not work properly when running 2 distinct instances of Microsoft Visual C++.NET in parallel.

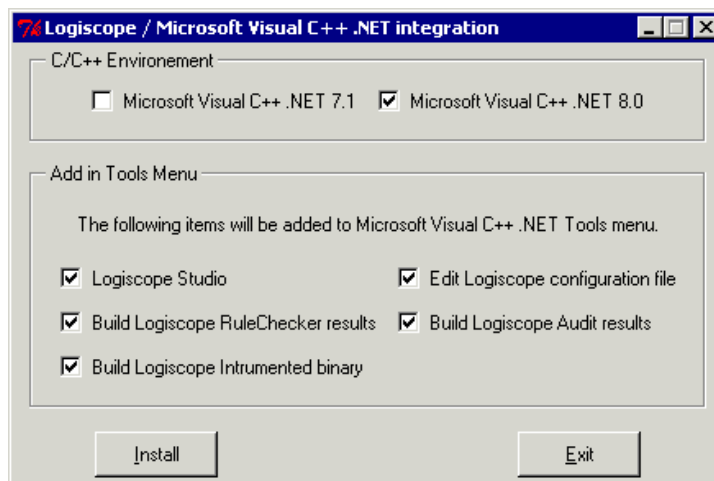
2. Installation

Before using the **Telelogic Logiscope Microsoft Visual C++ .NET** integration, you have first to install it.

1. Launch the installer from the Microsoft Windows Start menu: Program Files / Telelogic / Telelogic Lifecycle Solutions Tools / Telelogic Logiscope 6.5 / Visual C++ .NET Integration / Installation
2. One of the following windows appears:
 - If only one version of Visual C++ .NET is installed on the computer:



- If more than one version of Visual C++ .NET is installed, choose the Visual C++ .NET environments in which you want to install the integration:



3. Choose the options you want to install (by default everything is installed):
 - **Logiscope Studio** : if checked, the corresponding menu item “Logiscope Studio” will be added into the Tools menu of Visual C++ .NET, which will allow you to start Logiscope Studio from Visual C++ .NET environment.

- **Edit Logiscope configuration file** : if checked, the corresponding menu item will be added into the Tools menu of Visual C++ .NET, which will allow you to edit the configuration file directly within Visual C++ .NET. The configuration file is used to set default values which are used to create Logiscope projects.
- **Build Logiscope RuleChecker results** :if checked, the corresponding menu item will be added into the Tools menu of Visual C++ .NET, which will allow you to check rules on a Visual C++ project directly from Visual C++ .NET environment.
- **Build Logiscope QualityChecker results** : if checked, the corresponding menu item will be added into the Tools menu of Visual C++ .NET, which will allow you to make a quality analysis of a Visual C++ project directly from Visual C++ .NET environment.
- **Build Logiscope Instrumented binary** : if checked, the corresponding menu item will be added into the Tools menu of Visual C++ .NET, which will allow you to build an instrumented binary, for coverage analysis, from a Visual C++ project directly from the Visual C++ .NET environment.

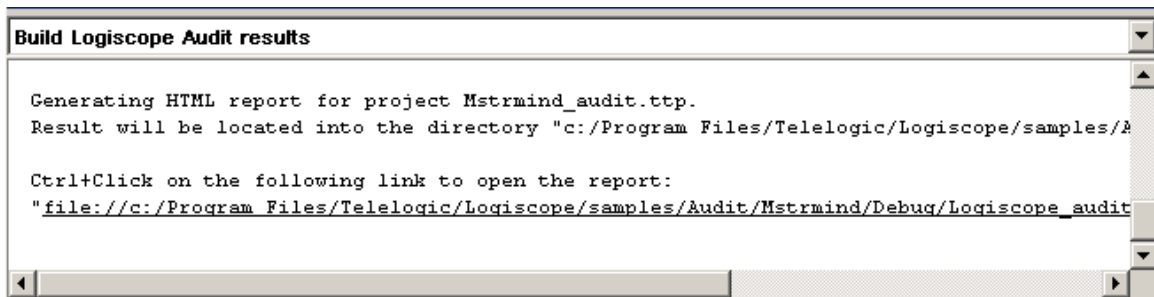
4. Click on Install for the integration to be installed.

3. Using Logiscope from Visual C++

3.1. Building Logiscope QualityChecker Results

Before using this functionality, the project must be cleaned in order to force a full rebuild, or new Configuration created for the project (to do so use the Configuration Manager of Microsoft Visual C++). The configuration may be called Logiscope.

Once this has been done, activate the menu “Tools/Build Logiscope QualityChecker Results” from Visual C++. The build starts and Logiscope is activated. Logiscope creates a new Logiscope project during the build and produces the *QualityChecker* results. Once the build is finished the following text appears within the output window of Visual C++:



```

Build Logiscope Audit results

Generating HTML report for project Mstrmind_audit.ttp.
Result will be located into the directory "c:/Program Files/Telelogic/Logiscope/samples/A

Ctrl+Click on the following link to open the report:
"file:///c:/Program Files/Telelogic/Logiscope/samples/Audit/Mstrmind/Debug/Logiscope_audit

```

The directory where the *QualityChecker* report has been generated is indicated, and a link to this report is available, using Ctrl+Click on the link will open the *QualityChecker* report of the project in a Microsoft Visual C++ .Net window.

During the build the following files/directories are created:

- *<project_name>_audit.ttp* located into the project directory is the Logiscope project corresponding to the Visual C++ project being processed.
- *<target_directory>/Logiscope_audit* is the Logiscope repository containing intermediate files produced by Logiscope.
- *<target_directory>/Logiscope_audit/audit_c* for C project or *<target_directory>/Logiscope_audit/audit_cpp* for C++ project is the Logiscope repository containing the HTML report.

3.2. Building Logiscope RuleChecker Results

Before using this functionality, the project must be cleaned in order to force a full rebuild, or new Configuration created for the project (to do so use the Configuration Manager of Microsoft Visual C++). The configuration may be called Logiscope.

Once this has been done, activate the menu “Tools/Build Logiscope RuleChecker Results” from Visual C++. The build starts and Logiscope is activated. Logiscope creates a new Logiscope project during the build and produces the RuleChecker results. Once the build is finished the following text appears in the output window of Visual C++:

```

Build Logiscope RuleChecker results
samples\Audit\Mstrmind\machine.c(189) : Portability_13_NoTab: Use of tabulation.
samples\Audit\Mstrmind\machine.c(190) : Portability_13_NoTab: Use of tabulation.
samples\Audit\Mstrmind\machine.c(191) : Portability_13_NoTab: Use of tabulation.
samples\Audit\Mstrmind\machine.c(192) : Portability_13_NoTab: Use of tabulation.
samples\Audit\Mstrmind\machine.c(196) : Portability_13_NoTab: Use of tabulation.
samples\Audit\Mstrmind\machine.c(204) : Portability_1_C++Keywords: C++ keywords not authc
samples\Audit\Mstrmind\machine.c(210) : Portability_13_NoTab: Use of tabulation.
samples\Audit\Mstrmind\machine.c(230) : Portability_13_NoTab: Use of tabulation.

```

Each violation of a rule is listed and a double click on it opens the source file on the appropriate line allowing immediate modification of the source file.

Scrolling down to the end of the output window will show the following text:

```

Build Logiscope RuleChecker results
C:\Program Files\Telelogic\Logiscope\samples\Audit\Mstrmind\machine.c(92) : Portability_1
C:\Program Files\Telelogic\Logiscope\samples\Audit\Mstrmind\machine.c(93) : Portability_1
C:\Program Files\Telelogic\Logiscope\samples\Audit\Mstrmind\machine.c(94) : Portability_1

Ctrl+Click on the following link to open the report:
"file:///c:/Program Files/Telelogic/Logiscope/samples/Audit/Mstrmind/Debug/Logiscope_rule/

```

A link to the rule violations report is available, using Ctrl+Click on the link will open the RuleChecker report of the project in a Microsoft Visual C++ .Net window.

During the build the following files/directories are created:

- *<project_name>_rule.ttp* located into the project directory is the Logiscope project corresponding to the Visual C++ project being processed.
- *<target_directory>/Logiscope_rule* is the Logiscope repository containing intermediate files produced by Logiscope.
- *<target_directory>/Logiscope_rule/rchk_c* for C project or *<target_directory>/Logiscope_rule/rchk_cpp* for C++ project is the Logiscope repository containing the html report.

3.3. Building Logiscope Instrumented binary

Before using this functionality, the project must be cleaned in order to force a full rebuild, or a new Configuration created for the project (to do so use the Configuration Manager of Microsoft Visual C++). The configuration may be called Logiscope.

Once this has been done, activate the menu “Tools/Build Logiscope Instrumented binary” from Visual C++. The build starts and Logiscope is activated. Logiscope creates a new Logiscope project during the build and produces an instrumented binary for coverage analysis.

During the build the following files/directories are created:

- *<project_name>_test.ttp* located in the project directory is the Logiscope project corresponding to the Visual C++ project being processed.
- *<target_directory>/<target_name>* is the instrumented binary build. Please refer to

TestChecker documentation for its usage.

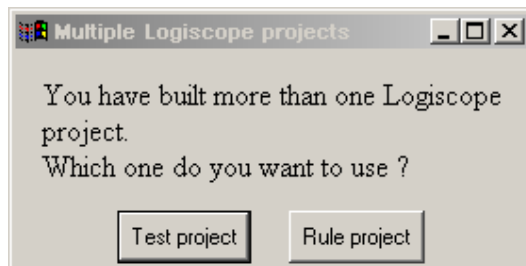
Note: The default instrumentation library which comes with Logiscope (`<install-dir>/instr/lib/vlgtc.lib`) is compiled using the Multi-Threaded model (the Single-Threaded model is no longer supported by Visual C++ .NET 2005). If the application to instrument should be a Single-Threaded one you would need to build a new library using the source file `<install-dir>/instr/src/vlgchk.c` and reference the newly built library using the configuration file (see 3.5 Edit Logiscope configuration file).

3.4. Launching Logiscope Studio

Once a Logiscope project has been built, a more comprehensive examination of Logiscope results may be achieved by launching Logiscope Studio. Logiscope Studio allows to go deeper in the exploitation of the produced Logiscope data and in turn to start Logiscope Viewer for a deep quality analysis or TestChecker for running tests and coverage analysis.

To start Logiscope Studio from Microsoft Visual C++ .NET graphical environment activate the menu “Tools/Logiscope Studio”.

If more than one Logiscope project have been built from the same Microsoft Visual C++ project, for instance a RuleChecker and a TestChecker project, a dialog pops up to ask which one has to be open. This dialog looks like:



Click on the kind of project you want to open with Logiscope Studio.

3.5. Customizing the integration

To customize the integration, you can edit the configuration file in Microsoft Visual C++ .NET environment.

Mainly, this file used to setting up:

- the C or C++ dialect for parsing code.
- the default quality model used by *Logiscope QualityChecker* for quality evaluation, default is the Logiscope.ref file delivered with the standard Logiscope distribution in the Ref folder.
- the default rule set(s) used by *Logiscope RuleChecker* for rule checking.
- the default support library used by *Logiscope TestChecker* to instrument source code under test.

```

#####
# This file is used for Logiscope / VC++ .NET
# integration customization.
#####

# Note:
# Into this script the tcl variable $InstallDir refers
# to the current Logiscope installation directory

#####
# Logiscope QualityChecker C
#####
# Dialect:
set CAuditDefaultDialect msc70

# Quality Model:
set CAuditDefaultQualityModelFile [file join $InstallDir Ref Logiscope.ref]

#####
# Logiscope QualityChecker C++
#####
# Dialect / parser configuration file:
set CppAuditDefaultParserConfigFile MFC

# Quality Model:
set CppAuditDefaultQualityModelFile [file join $InstallDir Ref Logiscope.ref]

#####
# Logiscope RuleChecker C
#####
# Dialect:
set CRuleDefaultDialect msc70

# Rule set: (Default is the Portability.rsf file from the installation)
set CRuleDefaultRuleSetFile [file join $InstallDir util Portability.rsf]

#####
# Logiscope RuleChecker C++
#####

```

Telelogic Logiscope

```
# Dialect / configuration file:
set CppRuleDefaultParserConfigFile MFC

# Rule set:
set CppRuleDefaultRuleConfigFile [file join $InstallDir data audit_c++
RuleChecker.cfg]

#####
# Logiscope TestChecker C
#####
# Dialect
set CTestDefaultDialect msc70

# Support instrumentation library: to be linked with the instrumented binary
set CTestDefaultInstrumentationLib [file join $InstallDir instr lib vlgtc.lib]

# Location of the prototypes of instrumentation functions
set CTestDefaultInclude "/I\"$::InstallDir/instr/include\" /I."

# set 0 or 1 to turn off or on MC/DC instrumentation
set CTestInstrumentMCDC 1

#####
# Logiscope TestChecker C++
#####
# Support instrumentation library: to be linked with the instrumented binary
set CppTestDefaultInstrumentationLib [file join $InstallDir instr lib
vlgtc.lib]

# Ignore file:
set CppTestDefaultIgnoreFile [file join $InstallDir util log_inst.ign]

# Alias file:
set CppTestDefaultAliasFile [file join $InstallDir util log_inst.al]

# Provides the tcl file to be used by the C++ instrumenter
set CppTestDefaultTclFile [file join $InstallDir util log_inst.tcl]

# Used during compilation of instrumented code to find prototypes of
# instrumentation functions
set CppTestDefaultInclude "/I\"$::InstallDir/instr/include\" /I."
```

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send written inquiries to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for

this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Copyright license

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, ibm.com, Telelogic, Telelogic Synergy, Telelogic Change, Telelogic DOORS, Telelogic Tau, Telelogic DocExpress, Telelogic Rhapsody, Telelogic Statemate, and Telelogic System Architect are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both, are trademarks of Telelogic, an IBM Company, in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at

www.ibm.com/legal/copytrade.html.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, FrameMaker, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions.

AIX and Informix are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

HP and HP-UX are registered trademarks of Hewlett-Packard Corporation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Macrovision and FLEXnet are registered trademarks or trademarks of Macrovision Corporation.

Microsoft, Visual, Windows, Windows 2003, Windows XP, Windows Vista and/or other Microsoft products referenced herein are either trademarks or registered trademarks of Microsoft Corporation.

Netscape and Netscape Enterprise Server are registered trademarks of Netscape Communications Corporation in the United States and other countries.

Sun, Sun Microsystems, Solaris, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Pentium is a trademark of Intel Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product or service names may be trademarks or service marks of others.