

IBM® Rational® Synergy for Microsoft Source Code Control
Interface—Visual C++

User's Guide



*IBM Rational Synergy for Microsoft
Source Code Control Interface—Visual C++*

Release 7.2

Before using this information, be sure to read the general information under Appendix, “Notices” on page 55.

This edition applies to **VERSION 7.2, IBM Rational Synergy for Microsoft Source Code Control Interface—Visual C++ (product number 5724V66)** and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1996, 2011**

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

Table of Contents

Chapter 1: Introduction	1
Product requirements	1
Assumptions.....	2
What Rational Synergy controls	2
Determining source code control options for controlled objects	3
Understanding how the interface works.....	3
Chapter 2: Installation Instructions	7
Before installing the interface	7
Installing the interface on your workstation.....	7
Customizing the Rational Synergy toolbar	8
Installing Microsoft file types	9
Chapter 3: Interface Strategies	11
Knowing the basics	11
Understanding Visual C++ dependent subprojects and Rational Synergy subprojects.....	12
Visual C++ subprojects	12
Rational Synergy subprojects	13
Compiling and linking moveable subprojects.....	14
Visual C++ dependent subprojects.....	14
Visual C++ reference subprojects.....	15
Determining which build strategy works best	16
Managing makefiles.....	16
Building from the command line	16
Using Rational Synergy timestamp options.....	17
Chapter 4: Data Migration	19
Migrating projects	19

Chapter 5: Project Setup	21
Getting started	21
Verifying project migration	22
Setting up for automated builds	23
Exporting makefiles when saving project changes	23
Setting the Rational Synergy timestamp option	23
Saving project changes before establishing baselines	24
Creating project baselines	24
Chapter 6: Daily Development	25
How developers use the interface	25
Updating changes in the Rational Synergy database	27
Making project configuration changes	27
Adding new projects to the workspace	27
Adding files, classes, or resources to projects	28
Removing files, classes, or resources from projects	28
Exporting makefile changes	29
Chapter 7: Build Management Process	31
How build managers use the interface	31
Copying build management projects	31
Reconfiguring prior to builds	32
Factors affecting the build process	32
Building in Visual C++ from the GUI	32
Without the timestamp option	32
With the timestamp option	33
Building in Visual C++ from the command line	33
Without setting the timestamp option	33
With setting the timestamp option	33

Chapter 8: Commonly Used Procedures

35

Copying projects	36
Checking out files	36
Single files	36
Multiple files	36
Checking out classes	37
Single classes	37
Multiple classes	37
Checking out resources	38
Single resources	38
Multiple resources	38
Opening a file, class, or resource	39
Checking in files, classes, and resources	39
Single files, classes, or resources	39
Multiple files, classes, or resources	40
Changing the properties of Rational Synergy files	40
Showing file histories	41
Obtaining latest versions of files, classes, and resources	41
Undoing check out	42
Adding uncontrolled files, classes, or resources	43
Deleting controlled files, classes, or resources	44
Closing projects	44
Displaying the Rational Synergy Work pane	45
Accessing task-based CM features	45
CMS Bar toolbar	45
Synergy advanced operations dialog box	46
Options dialog box	46
Work Area dialog box	47
Task-Based CM dialog box	48
Selecting tasks	49
Creating tasks	49
Completing tasks	50
Updating projects	50

Syncing projects	50
Working in disconnect mode	51
Disconnecting	51
Reconnecting	51
Removing projects from source code control	52
Subproject control	52
Parallel version control	52
Default check out and check in comments	53
Appendix: Notices	55
Trademarks	57
Index	59

1

Introduction

The Rational® Synergy for Microsoft® Source Code Control Interface™ allows you to perform Rational® Synergy™ source code control operations without leaving the Microsoft® Visual C++® environment.

You can use the Visual C++ built-in source code control menus and dialog boxes to perform the most common source code control operations. The interface supports common operations such as checking in and checking out, as well as task-based operations such as selecting and completing tasks. The interface also provides you with easy access to the Rational Synergy interface so you can expand your change management capabilities with other change management tools and features.

Also, the Rational Synergy for Microsoft Source Code Control Interface provides support for multiple developers working on the same project. All Rational Synergy objects are stored in a database, and each checked-in object can be made available for testing or general use based on the criteria you establish when you start working on a project. Therefore, all developers on a project have immediate access to files, classes, and resources that have been developed, tested, and checked in by others.

Product requirements

The following products are required to run the Rational Synergy for Microsoft Source Code Control Interface:

- Visual C++ 6.0
- Rational Synergy 7.2
- Rational Synergy for Microsoft Source Code Control Interface 7.2, including Rational Synergy Microsoft Developer types

Assumptions

This Rational Synergy for Microsoft Source Code Control Interface documentation assumes that you have a basic understanding of how to use Visual C++ and Rational Synergy. Information about Rational Synergy in this document is limited to information that is considered essential to your use of the interface. If you need additional information about Rational Synergy, the following Rational documentation is available on the [Synergy Information Center site](http://publib.boulder.ibm.com/infocenter/synhelp/v7m2r0/index.jsp), at <http://publib.boulder.ibm.com/infocenter/synhelp/v7m2r0/index.jsp>.

- For terms, concepts, and methodology, see *Introduction to Rational Synergy*.
- For information about build management level-operations, see the *Rational Synergy Build Manager's Guide*.
- For a description of dialogs, commands, and default settings, refer to the Rational Synergy Help for the specific dialog, command, or default setting for which you need help.

Read this document in its entirety before you migrate data and set up your projects. You will need to make several key decisions based on the type of projects you are using, the version of Visual C++ you are running, the way you do your builds, and whether you are using the timestamp option. Taking the time now to evaluate these factors is beneficial for a seamless, efficient interface.

What Rational Synergy controls

Visual C++ represents the objects in your project as files, classes, and resources. In reality, these objects consist of files that are parsed by Visual C++ to determine whether they belong in a file, class, or resource group. Visual C++ automatically maintains this information.

Rational Synergy controls the files (such as .cpp, .c, and .exe) that comprise file, class, and resource groups. Rational Synergy also controls the files (such as .dsw and .dsp) that Visual C++ maintains.

Determining source code control options for controlled objects

To determine whether a file, class, or resource is under source code control, right-click the object to display its menu. If the object is controlled, the pop-up menu contains source code control options for a controlled object. Otherwise, the menu only contains the option to add it to source code control.

File objects provide an additional visual cue, as illustrated below:

If the file is controlled, its icon has a grey background. 

If the file is not controlled, its icon has a white background. 

Class and resource objects have no visual cue indicating whether or not they are controlled.

The source code control options available in an object pop-up menu depend on whether the object is a file, class, or resource, and whether its associated files are checked out or read-only. File and class objects provide a visual cue that indicates whether they are checked out or read-only. A checked-out file has a check mark next to the file icon. Resource objects have no visual cue indicating whether they are checked out or read-only.

Understanding how the interface works

The Rational Synergy for Microsoft Source Code Control Interface uses the Microsoft Source Code Control Interface (MSSCCI) Microsoft Application Programming Interface (MAPI) to provide source code control to Visual C++. The Rational Synergy dynamic link library (DLL) file, `ccm_scc.dll`, supports the interface.

Version control menu items, such as **Check In** and **Check Out**, are built into Visual C++ and are available on the **Source Control** toolbar. Each requested operation is handled in the foreground by Visual C++. Rational Synergy operates in the background. More advanced operations are available outside of the MAPI, from the Rational Synergy interface.

The Visual C++ source code control operations are context sensitive. Menu options are available only when they are appropriate for the type and state of the file, class, or resource on which you are performing the operation. Otherwise, the options are greyed-out.

Contacting IBM Rational Software Support

If the self-help resources have not provided a resolution to your problem, you can contact IBM® Rational® Software Support for assistance in resolving product issues.

Note If you are a heritage Telelogic customer, a single reference site for all support resources is located at <http://www.ibm.com/software/rational/support/telelogic/>

Prerequisites

To submit your problem to IBM Rational Software Support, you must have an active Passport Advantage® software maintenance agreement. Passport Advantage is the IBM comprehensive software licensing and software maintenance (product upgrades and technical support) offering. You can enroll online in Passport Advantage from <http://www.ibm.com/software/lotus/passportadvantage/howtoenroll.html>

- To learn more about Passport Advantage, visit the Passport Advantage FAQs at http://www.ibm.com/software/lotus/passportadvantage/brochures_faqs_quickguides.html.
- For further assistance, contact your IBM representative.

To submit your problem online (from the IBM web site) to IBM Rational Software Support, you must additionally:

- Be a registered user on the IBM Rational Software Support web site. For details about registering, go to <http://www.ibm.com/software/support/>.
- Be listed as an authorized caller in the service request tool.

Submitting problems

To submit your problem to IBM Rational Software Support:

1. Determine the business impact of your problem. When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting.

Use the following table to determine the severity level.

Severity	Description
1	The problem has a <i>critical</i> business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
2	This problem has a <i>significant</i> business impact: The program is usable, but it is severely limited.
3	The problem has <i>some</i> business impact: The program is usable, but less significant features (not critical to operations) are unavailable.
4	The problem has <i>minimal</i> business impact: The problem causes little impact on operations or a reasonable circumvention to the problem was implemented.

2. Describe your problem and gather background information. When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Rational Software Support specialists 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?
To determine the exact product name and version, use the option applicable to you:
 - Start the IBM Installation Manager and select **File > View Installed Packages**. Expand a package group and select a package to see the package name and version number.
 - Start your product, and click **Help > About** to see the offering name and version number.

- What is your operating system and version number (including any service packs or patches)?
 - Do you have logs, traces, and messages that are related to the problem symptoms?
 - Can you recreate the problem? If so, what steps do you perform to recreate the problem?
 - Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, or other system components?
 - Are you currently using a workaround for the problem? If so, be prepared to describe the workaround when you report the problem.
3. Submit your problem to IBM Rational Software Support. You can submit your problem to IBM Rational Software Support in the following ways:
- **Online:** Go to the IBM Rational Software Support web site at <https://www.ibm.com/software/rational/support/> and in the Rational support task navigator, click **Open Service Request**. Select the electronic problem reporting tool, and open a Problem Management Record (PMR), describing the problem accurately in your own words.

For more information about opening a service request, go to <http://www.ibm.com/software/support/help.html>
 - You can also open an online service request using the IBM Support Assistant. For more information, go to <http://www.ibm.com/software/support/isa/faq.html>.
 - **By phone:** For the phone number to call in your country or region, go to the IBM directory of worldwide contacts at <http://www.ibm.com/planetwide/> and click the name of your country or geographic region.
 - **Through your IBM Representative:** If you cannot access IBM Rational Software Support online or by phone, contact your IBM Representative. If necessary, your IBM Representative can open a service request for you. You can find complete contact information for each country at <http://www.ibm.com/planetwide/>.

2

Installation Instructions

Before installing the interface

Before you install the Rational Synergy for Microsoft Source Code Control Interface, verify that the following programs are installed and are operational:

- Visual C++ 6.0
- Rational Synergy 7.2

Installing the interface on your workstation

To install the interface on your workstation:

4. Double click the `setup.exe` file.
The **Welcome** dialog box appears.
5. Click **Next**.
The **License Agreement** dialog box appears.
6. Read and accept the terms in the license agreement, and then click **Next**.
The **Destination Folder** dialog box appears.
7. Verify the installation path, and then, if the path is correct, click **Next**.
If you need to change the path:
 - a. Click **Change**.
The **Change Current Destination Folder** dialog box appears.
 - b. In **Look in**, navigate to and select the correct path, and click **OK** to close the **Change Current Destination Folder** dialog box.
 - c. Click **Next**.
The **Ready to Install the Program** dialog box appears.
8. Click **Install**.
9. When the **InstallShield Wizard Completed** dialog box appears, click **Finish**.
10. Start a Rational Synergy session on your production database.

Customizing the Rational Synergy toolbar

After Rational Synergy has been installed, you can dock the Rational Synergy **CMS Bar** toolbar (see “CMS Bar toolbar” on page 45) to the Visual C++ toolbar area.

To add the toolbar:

1. Start Visual C++, and do not open any workspaces.
2. From the main menu, select **Tools > Customize** or right-click the Visual C++ toolbar area and select **Customize**.

The **Customize** dialog box appears.

3. Click the **Add-ins and Macro Files** tab.
4. Deselect **CMSynergyBar**, and then close the dialog box.
5. Reopen the dialog box, and then select **CMSynergyBar**.
6. Click the **Toolbars** tab.
7. In the **Toolbars** section, select **CMS Bar**.

Installing Microsoft file types

Microsoft file types must be installed into each Rational Synergy database before you can use the interface. Your group membership must include *ccm_admin* privileges.

To install Microsoft file types:

1. Start the Rational Synergy Client.
2. From the main menu, select **Tools > Start Synergy CLI**
3. Set your role to *ccm_admin* by running the command `ccm set role ccm_admin`
4. From the command prompt, run the `mstypes_install.bat` file, which is located by default in `C:\Program Files\IBM\Rational\Synergy for MSSCCI\bin`.
5. After the installation is complete, exit and restart your Rational Synergy session.

3

Interface Strategies

Knowing the basics

This section provides important information to consider before migrating data into Rational Synergy. It is important to understand how projects are structured in both Visual C++ and Rational Synergy. Both products use the subproject concept, but the two products do not treat subproject dependencies in the same manner. Detailed information is provided to help you understand these interactions.

Additional information is presented so that you can determine the best strategy to use when migrating data. Several factors affect how projects should be set up prior to migration. Having the necessary information before starting the migration allows you to make the proper choices when structuring your data. Understanding your project structure and how it will be treated after it is imported into Rational Synergy allows your migration to proceed smoothly. It is easier to make changes to the project structure in Visual C++ before migrating data into Rational Synergy.

Read the following sections to help you assess your current project structure and to determine how the information will be treated after migration:

- “Understanding Visual C++ dependent subprojects and Rational Synergy subprojects” on page 12
- “Compiling and linking moveable subprojects” on page 14
- “Determining which build strategy works best” on page 16

Understanding Visual C++ dependent subprojects and Rational Synergy subprojects

Both Visual C++ and Rational Synergy use the concept of a subproject. However, subprojects and their dependencies based on location are not treated in the same manner, which can lead to confusion.

Visual C++ subprojects

A Visual C++ subproject is used to define dependencies between components of an application. The table below shows an example of an application, its dependent libraries, their locations, and their build targets.

Project	Location	Module
MYAPP	C:\myapp	MYAPP.EXE
CONTROLS	C:\myapp\controls	CONTROLS.LIB
GUI	C:\myapp\gui	GUI.LIB
OSINTF	D:\common\osintf	OSINTF.LIB

Visual C++ uses a workspace to define a project and its dependencies. A workspace is defined in the `.dsw` file. This file contains dependency information in addition to source code control information. The build information for a project is contained in a `.dsp` file. In Visual C++, each project and subproject has its own `.dsp` file. The Visual C++ build engine uses this dependency information to check each subproject before building the **MYAPP** project. There are no restrictions on the location of Visual C++ subprojects. They may be relative to the parent project or located in some other absolute location.

However, the `.dsp` base filename must match the name of the Rational Synergy project. For example, you cannot create a `.dsp` file named `foo.dsp` in a Rational Synergy subproject named `foo_ext`. You must create the `.dsp` as `foo_ext.dsp`.

The `.dsp` base filename must match the name of the component you are building if you want to include that `.dsp` as a dependency of another `.dsp`. For example, if you add the Visual C++ subproject **bar** as a dependency in the subproject **foo**, linking `foo.exe` automatically includes (and is dependent on) `bar.lib`. If you change the Visual C++ project settings for **bar** so that it generates `abc.lib` instead of `bar.lib`, then the Visual C++ link does not include this file.

Rational Synergy subprojects

A Rational Synergy subproject is also used to define dependencies between components. These dependencies define the project configuration. The table below defines an application, its dependent subprojects, and their current locations. The build targets are the same as in the table in the section, “Visual C++ subprojects” on page 12.

Project	Object type	Location
MYAPP	Project	C:\ccm_wa\myapp-1\myapp
CONTROLS	Project	C:\ccm_wa\myapp-1\myapp\controls
GUI	Project	C:\ccm_wa\myapp-1\myapp\gui
OSINTF	Project	D:\ccm_wa\osintf-1\osintf

A Rational Synergy subproject may have an absolute path, as in **OSINTF**, or a relative path, as in **CONTROLS** and **GUI**. An important aspect of absolute Rational Synergy subprojects is that their locations may change when the project is updated. This occurs because the version string is embedded in the path. A Rational Synergy use or update operation can cause new subproject versions to be assigned to the main project. For example, when version 2 of the **OSINTF** subproject is required for the **MYAPP** project, its location may change to D:\ccm_wa\osintf-2\osintf, or some other location.

This important issue causes difficulties in trying to maintain the relationship between the workspace project definitions and those of the Rational Synergy subprojects. Specifically, these are the issues:

- The Rational Synergy project path is stored in source code control information in the .dsw file.
- The Rational Synergy project path is stored in the dependency information of the main project .dsp file.

A Visual C++ subproject does not have to correspond directly to a Rational Synergy subproject. The table below shows a sample configuration.

Project	Object type	Location
MYAPP	Project	C:\ccm_wa\myapp-1\myapp
CONTROLS	Directory	C:\ccm_wa\myapp-1\myapp\controls
GUI	Directory	C:\ccm_wa\myapp-1\myapp\gui
OSINTF	Project	D:\ccm_wa\osintf-1\osintf

The difference is that the **CONTROLS** and **GUI** objects are directories, not projects. The settings in the Visual C++ subproject structure do not need to change, see “Visual C++ subprojects” on page 12. This type of project setup does not cause the same difficulties as described earlier. In essence, a Rational Synergy relative subproject and a directory may both be represented by the Visual C++ .dsp file without any naming issues.

Another issue with Visual C++ subprojects occurs within dynamic software development teams when project changes are required. When two developers modify the Visual C++ configuration by adding different subproject dependencies, these changes need to be merged and then propagated back to all developers.

Compiling and linking moveable subprojects

Another issue to consider is effectively building with Rational Synergy subprojects that may move during **update** operations. The location of the Visual C++ dependent and Visual C++ reference subprojects are two areas of concern.

Visual C++ dependent subprojects

A Visual C++ dependent subproject is part of the Visual C++ workspace. This project is rebuilt when the Visual C++ build engine detects changes. The main project will be subsequently rebuilt after the dependent project is updated. In this case, Visual C++ stores the absolute location of the dependent subproject when the project is added to the workspace. If the project location changes, you need to manually edit the work space to redirect the location.

Visual C++ reference subprojects

A Visual C++ reference subproject is one that does not contain a corresponding DSP, but is needed during the compile and link processes. For example, the table below shows the addition of a third party library, TPLIB.LIB, to the **MYAPP** project in the Rational Synergy project configuration.

Project	Object type	Location
MYAPP	Project	C:\ccm_wa\myapp-1\myapp
CONTROLS	Project	C:\ccm_wa\myapp-1\myapp\controls
GUI	Project	C:\ccm_wa\myapp-1\myapp\gui
OSINTF	Project	D:\ccm_wa\osintf-1\osintf
TPLIB	Project	E:\ccm_wa\tplib-1\tplib

Since this library is only provided in object form, no build is required. To access the header files during compilation and the library file during linking, one of two methods may be used.

Ideally, this library would be added as an object to the main project. Alternatively, the Visual C++ project settings could be modified using the **C++** and **Link** tabs by placing references to the absolute path (i.e., E:\ccm_wa\tplib-1\tplib) in the appropriate fields. The first method is preferred since it allows the Visual C++ build engine to track updates to the new library and to rebuild the executable as necessary.

In either case, the new header and library files are accessed through the absolute path provided. If the project location changes over time, these settings may have to be changed manually.

Determining which build strategy works best

Visual C++ automatically creates and maintains the internal build files (.dsp files). However, these files are not compatible with NMAKE. Two options are available to automate your builds:

- Managing the makefile (.mak file)
- Building from the command line

The following sections discuss how each option affects the development and build management processes.

Managing makefiles

Visual C++ can be configured either to allow everyone to create and manage the makefile, or to rely on the build manager to create and maintain the makefile.

To have all project changes reflected to the .mak file, the Visual C++ project can be configured to always export the makefile when saving project settings. If you choose this method, all users of the interface are required to manage the .mak file using the Rational Synergy Check-Out/Check-in commands or with the Rational Synergy sync command because Visual C++ does not automatically check out the makefile when it needs to write to it.

If only the build manager creates and maintains the makefile, the interface provides a macro for use in a build manager script that automates the makefile export process. When using this method, the developers can always perform their builds within the Visual C++ environment. There is no need for makefile maintenance.

See “Setting up for automated builds” on page 23 for examples of using both options.

Building from the command line

You can build your project from the command line using the information stored in the Visual C++ project (.dsp file).

Note If you are using Visual C++ 6.0, these macros can be replaced by the command line build capabilities of the msdev command.

For additional information about this feature, use the Visual C++ Help by selecting **Help > Search**, and then typing **“Building a project from the command line”** (include the quotes).

Using Rational Synergy timestamp options

Rational Synergy provides an option for maintaining the timestamps of files in a Rational Synergy work area. If you use this option, the timestamp of a file in the build manager's work area is the same as the time the file was copied into the work area. This causes the file to appear newer than its corresponding object file. The make tool then determines that the object file is out of date with respect to its source, and the tool attempts to rebuild it.

If you do not set this option, Rational Synergy does not update the modification time of the file when a different version replaces the original version. When you are using the Rational Synergy interface instead of Visual C++, it is possible to replace the current version with a version that has an older timestamp. If the product depends on this file and the product has a timestamp newer than the timestamp of the file, the Visual C++ build fails to rebuild the product. This occurs because Visual C++ is not able to correctly identify the changed version.

For this reason, if you are changing your project configuration from the Rational Synergy for Microsoft Source Code Control Interface, you must either immediately rebuild each newly used or replaced object version in your project, or you must regularly force a rebuild of the entire project to ensure that all new object versions are built, regardless of their timestamps.

For more information about timestamps, see “Setting the Rational Synergy timestamp option” on page 23 and “Factors affecting the build process” on page 32.

4

Data Migration

This section describes the steps required to migrate an entire Visual C++ application into Rational Synergy. The information presented here assumes that you have read the “Interface Strategies” chapter of this document, and that your project structure is set up for use with the Rational Synergy for Microsoft Source Code Control Interface.

Migrating projects

Before you can use source code control operations on the files, classes, and resources in your Visual C++ project, you or another user (such as the CM administrator or the build manager) must migrate the project into the Rational Synergy database.

To migrate a project into the Rational Synergy database:

1. If the Visual C++ file types and migrate rules are not installed already, install them.

If you have followed the instructions in the “Installation Instructions” chapter of this document, you have already installed the types.

2. If the project that you want to migrate is open in Visual C++, close it.
3. Start the Rational Synergy Client.
4. Create a project in Synergy and specify the workarea as the location where your Visual C++ project is located.
5. Right click on the project and choose **Sync Work Area > Members Only**
6. Select all the files you wish to migrate to Synergy.
7. Right click and choose **Keep Work Area Changes**
8. Confirm that the .dsw and .dsp files for your Visual Studio project are located in the root directory of your Synergy project. The integration requires that these files be in the root directory to function properly.

5

Project Setup

Getting started

This section describes what the build manager needs to do after migrating data into Rational Synergy (see the “Data Migration” chapter). Customizing the data while setting up the project allows you to get the most out of the Rational Synergy for Microsoft Source Code Control Interface.

The information presented here assumes that you have read the “Interface Strategies” chapter, successfully migrated your data using the information provided in the “Data Migration” chapter, and are ready to run the interface.

The following information is discussed in this section:

- “Verifying project migration” on page 22
- “Setting up for automated builds” on page 23
- “Setting the Rational Synergy timestamp option” on page 23
- “Saving project changes before establishing baselines” on page 24
- “Creating project baselines” on page 24

Verifying project migration

To verify the product migration:

1. In the Rational Synergy **Work** pane, double-click the `.dsw` file to start Visual C++.

Use this method to start the Rational Synergy for Microsoft Source Code Control Interface every time you run the interface. This method ensures that the proper relationships between subprojects are maintained, and that the environment variables are set correctly.

2. In Visual C++, when prompted to check out the `.dsp` file, click **Yes**.
The **Check out file(s)** dialog box appears.

3. In **Comment**, type your comment, and then click **OK**.

4. From the Visual C++ **File View**, right-click the workspace file, and then select **Add to Source Control**.

A message appears stating that a project with the same name already exists.

5. Click **Yes** to reconnect to the existing project.
6. In the Visual C++ **File View**, right-click the workspace file, and then select **Check Out**.

The **Check out file(s)** dialog box appears.

7. In **Comment**, type your comment, and then click **OK**.

At this point, your project has been migrated and is under source control. The workspace and project files have been checked out, which is required because they will be modified during this project setup process.

You can verify that your project is under Rational Synergy control by displaying the **Project Workspace File** pane and examining the icons. As discussed in “Determining source code control options for controlled objects” on page 3, controlled files have grey backgrounds. Uncontrolled files have white backgrounds.

Setting up for automated builds

This section discusses the two options that are available to automate your builds, which are described in “Determining which build strategy works best” on page 16.

Exporting makefiles when saving project changes

To set up a project to always export the makefile when saving project changes, each user should follow these steps to run Visual C++:

1. From the Visual C++ main menu, select **Tools > Options**, and then click the **Build** tab.
2. Select the **Export makefile when saving project file** check box.
3. Click **OK**.

Setting the Rational Synergy timestamp option

When a project is created, the default setting for the work area file timestamps is to maintain the modification time of each file as in prior releases. If you do not want to use the default setting, change it now.

To set the work area file timestamp option so that the timestamp on a file is updated as it is copied into the work area, use the following command:

```
ccm work_area -wa_time on project-version
```

The following examples show all options for this command:

```
ccm work_area -wa_time on myapp-1  
Project 'myapp-1' will now use work area times
```

```
ccm work_area -wa_time show myapp-1  
Project 'myapp-1' is using work area times
```

```
ccm work_area -wa_time off myapp-1  
Project 'myapp-1' will no longer use work area times
```

Note When a project is checked out, the project inherits the same timestamp setting as its baseline.

Saving project changes before establishing baselines

Before establishing your baseline, save the changes that you have just made to your project.

To save your changes:

1. Exit from your workspace.

A prompt appears asking if you want to save the changes to your project.

2. Click **Yes**.

Creating project baselines

After your data is successfully migrated and the initial project configuration is completed, you need to baseline your project. Creating a baseline allows you to save a copy of this build for future reference. When you create the baseline, you can choose to publish it and make changes available to developers immediately or wait to publish the baseline until the build has had more testing.

For information about creating baselines, see the baseline instructions in the [*Rational Synergy Build Manager's Guide*](#).

6

Daily Development

How developers use the interface

The Rational Synergy for Microsoft Source Code Control Interface allows you to use task-based methodology to work in Visual C++. You can begin work once the development project hierarchy is set up in Visual C++. From the Visual C++ **Project** menu, you can perform many source control operations, such as:

- Checking in and checking out a file, class, or resource
- Undoing a check out

Before you begin work on a project, you must check out your own development project in Rational Synergy. This operation is performed only once per project. See “Copying projects” on page 36 for the step-by-step instructions.

The following steps illustrate a typical development cycle. These steps guarantee a proper environment for your development efforts.

1. Update your Rational Synergy project.
 - a. In the first line in the Rational Synergy **Work** pane, right-click the project object.
 - b. Select either **Update > Members Only** or **Update > Members and Subprojects**.

This updates your project work area to contain the latest approved changes in your development cycle. **This operation should be performed with caution.** If you are working on a long development task, say one lasting several days, a update could bring in other developer changes that may have significant impact on your current development work.

See the Rational Synergy Help for more information.

2. Start Visual C++ by double-clicking the .dsw object in the Rational Synergy **Work** pane.
3. Minimize your Rational Synergy **Work** pane.

Do this after your workspace is opened in Visual C++, as you will be working in Visual C++. This ensures that a minimum number of dialogs are displayed while you work.

4. Click  to create a task or click  to select a task, and then click **OK**.

See “Creating tasks” on page 49 or “Selecting tasks” on page 49.

All Rational Synergy operations should begin with selecting or creating a task. This task provides a context in which you make changes and is used to associate source objects. **Always select a task before checking out files and making changes.**

You can select only one task per Rational Synergy session. If you are running multiple Visual C++ sessions, they share the same task. To avoid associating changes with the wrong task, run one Visual C++ session at a time.

5. Check out objects (classes, resources, or files) to modify. The objects are automatically associated with the task you have selected.

For more specific information about checking out objects, see “Commonly Used Procedures” on page 35.


6. Add new objects.

For more specific information about adding new objects, see “Adding files, classes, or resources to projects” on page 28.

If you add new source objects during your development, for example, in the Classic Wizard, Visual C++ automatically prompts you to add these objects to source control.

7. Make your changes (follow your code/debug cycle) to the checked out objects.

8. Save all files.

9. Click  to complete your task.

For more specific information about completing tasks, see “Completing tasks” on page 50.


When you complete the task, you are also checking in all files associated with the task. When you click **OK**, all of the files shown are checked in, and then the task is checked in.

10. From the Visual C++ main menu, select **Project > Source Control > Refresh Status** to display the most current data.

Updating changes in the Rational Synergy database

When using the Rational Synergy **Work** pane to edit your checked-out objects, Rational Synergy updates the database with the changes made to your work area files. When using the interface to check out and edit your files, the work area files are not automatically synced with the Rational Synergy database.

If you have files checked out for any length of time, it is important to sync your changes to the Rational Synergy database. This provides a way to back up work that is in process.

To sync your changes, in Visual C++, click , and then use the **Synergy Sync** dialog box to push the work area changes up to the Rational Synergy database.

You can also use the **Work Area** tab in the **Synergy** advanced operations dialog box to sync your work area. See “Work Area dialog box” on page 47.

Making project configuration changes

Several types of changes may require that you make modifications to the project configuration and build files (.dsw, .dsp, and .mak). The following sections describe some common configuration changes and the steps to take to accomplish the changes with the Rational Synergy for Microsoft Source Code Control Interface.

Adding new projects to the workspace

You can make changes to your workspace, such as adding other projects or dependent libraries. These changes must be reflected in your main .dsw module. When using the standard Visual C++ launching technique, a copy of the main .dsw module is used. This file is not controlled as part of your Rational Synergy project, so changes to it are not propagated to your team.

To make permanent changes to your workspace:

1. In the Rational Synergy **Work** pane, select your task.
2. Right-click the .dsw file, and then click **Check Out**.
3. Double-click the new .dsw version to launch Visual C++.
4. Make your configuration changes.
5. In the Visual C++ **FileView**, right-click the .dsw file (workspace object), and then click **Check In**.



The **Check in file(s)** dialog box appears.

6. In **Comment**, type your comment, and then click **OK**.

Adding files, classes, or resources to projects

During the normal course of development, you constantly add new objects to your Visual C++ project. Visual C++ reminds you when changes need to be made to the .dsp file. The system also reminds you that the file is not checked out. Additionally, Visual C++ asks you to put new objects under source control.

To add a new class to your project:

1. In Visual C++, click  to select your task.
See “Selecting tasks” on page 49.
2. From the main menu, select **Insert > New Class**.
The **New Class** dialog box appears.
3. In **Class information**, type your class information, and then click **OK**.
If your .dsp file is not checked out, a message appears asking if you want to put the newly added file(s) under source code control.
4. Click **Yes**.
The **Add to Source Control** dialog box appears.
5. In **Comment**, type your comment, and then click **OK**.
A message appears asking if you want to put the newly added file(s) under source code control.
6. Click **OK**.
7. Make the necessary code changes.
8. Click  to complete your task.
This method automatically checks the .dsp file and directory objects into the Rational Synergy database, which does not happen when you individually check in the files.

Removing files, classes, or resources from projects

For detailed information about removing files, classes, or resources from your project, see “Deleting controlled files, classes, or resources” on page 44.




Exporting makefile changes

If your environment requires the exporting of .mak files for build management, extra steps need to be taken when adding or removing files from your project because Visual C++ does not automatically check out a .mak file.

If your system is set up to automatically export a makefile when making project changes, you must check out the .mak file from the Rational Synergy **Work** pane before checking in or saving your project file.

If you export the .mak file using **Project > Export Makefile**, you may perform this operation any time after saving the project changes.

To export a makefile:

1. In Visual C++, click  to select a task.
See “Selecting tasks” on page 49.
2. From the main menu, select **Project > Export Makefile**.
The **Export Makefile(s)** dialog box appears.
3. Click **OK**.
4. Click  to sync your work area.
5. Click  to complete your task and to check in your changes.

Failure to check out the .mak file prior to regenerating it results in the following error:

```
<project name>.mak  
Error opening project file for save.  
Access denied.
```


7

Build Management Process

How build managers use the interface

This section contains information about what to do before you build and how to use Rational Synergy for Microsoft Source Code Control Interface to build controlled projects. Because many possible scenarios exist, depending on the environment in which you are working, there is not a correct or best way to do a build.

To read about the different possibilities and to obtain specific build instructions for several different scenarios, see “Factors affecting the build process” on page 32. To find the procedure tailored for you, match the information presented here to the environment in which you are working.

Copying build management projects


Before you begin work on a project, you must copy your build management project in Rational Synergy. This operation generally needs to be performed once per release.

To copy a build management project, see the “Copy a Project” section in the Rational Synergy Help.

You may need to repeat this operation for each stage of your software development lifecycle, each time using a different **Purpose** value.

For more detailed information about this process and about establishing and maintaining folders for task-based CM, see the [Rational Synergy Build Manager's Guide](#).

Reconfiguring prior to builds

Before you start your build in Visual C++, click  to update your project in Rational Synergy. Doing so updates your project with the most valid information. For complete information about the **Update** feature, see the Rational Synergy Help.

Also, make sure that you understand how to effectively build with Rational Synergy subprojects that may move during the update operations. The two areas of concern are the location of the following subprojects:

- Visual C++ dependent
- Visual C++ reference

For detailed information about building with moveable subprojects, see “Compiling and linking moveable subprojects” on page 14.

Factors affecting the build process

The process for running the build varies, depending on how your projects are set up and what versions of software you are running. Specific information is given for different build processes, depending on whether you are using any of the following:

- Visual C++ to build
- Timestamp option
- Projects using subprojects
- GUI or command line

Building in Visual C++ from the GUI

From the GUI in Visual C++, you can build without or with the timestamp option.

Without the timestamp option

To build without setting the timestamp option:

1. In the Rational Synergy **Work** pane, double-click the `.dsw` file to launch Visual C++.
2. Select **Build > Batch Build**.
3. In **Project configurations**, select the configuration(s) you wish to build, and then click **Rebuild All**.

With the timestamp option

To build with the timestamp option:

1. In the Rational Synergy **Work** pane, double-click the .dsw file to launch Visual Studio.
2. From the Visual C++ main menu, select **Build > Batch Build**.
3. In **Project configurations**, select the configuration(s) you wish to build, and then click **Build**.

Building in Visual C++ from the command line

In Visual Studio 6.0, you can execute builds from the command line using the /MAKE syntax.

For example, the following command builds the debug version of **MyApp**:

```
msdev MyApp.dsw /MAKE 'MyApp - Win32 Debug' /REBUILD
```

For further information about the available options, consult the Visual Studio documentation or type `msdev /?` from the command line.

Without setting the timestamp option

Do a rebuild using the following command:

```
msdev myapp.dsw /MAKE "MyApp - Win32 Debug" /REBUILD
```

With setting the timestamp option

Using the timestamp option allows you to perform an incremental build, since the timestamp of the object is updated when the object is copied to the build manager folder. To perform an incremental rebuild, use the following command:

```
msdev myapp.dsw /MAKE "MyApp - Win32 Debug"
```


8

Commonly Used Procedures

The following procedures are described in this section:

- “Copying projects” on page 36
- “Checking out files” on page 36
- “Checking out classes” on page 37
- “Checking out resources” on page 38
- “Opening a file, class, or resource” on page 39
- “Checking in files, classes, and resources” on page 39
- “Changing the properties of Rational Synergy files” on page 40
- “Showing file histories” on page 41
- “Obtaining latest versions of files, classes, and resources” on page 41
- “Undoing check out” on page 42
- “Adding uncontrolled files, classes, or resources” on page 43
- “Deleting controlled files, classes, or resources” on page 44
- “Closing projects” on page 44
- “Displaying the Rational Synergy Work pane” on page 45
- “Accessing task-based CM features” on page 45
- “Selecting tasks” on page 49
- “Creating tasks” on page 49
- “Completing tasks” on page 50
- “Updating projects” on page 50
- “Syncing projects” on page 50
- “Working in disconnect mode” on page 51
- “Removing projects from source code control” on page 52
- “Subproject control” on page 52
- “Parallel version control” on page 52
- “Default check out and check in comments” on page 53

Copying projects

To use Visual C++ with a Rational Synergy controlled project, you must start with a writable version of the project. The project must be in a writable state before you can make changes to any objects in the project.

Before beginning this procedure, obtain from your build manager the baseline project name and version.

To copy a project in Rational Synergy, see the “Copy a Project” section in the Rational Synergy Help.

When the copy project process is complete, the new project and files display in the Rational Synergy **Work** pane.

Checking out files

The single read-only file check-out is automatically initiated when you change the file. To check out multiple files, from the Visual C++ main menu, select **Project > Source Control > Check Out**.

Note Ensure that you have selected the appropriate task before checking out any objects. Any object you check out is associated with the selected task.

Single files

To check out a single file:

1. Right-click a read-only file, and then click **Check Out**.

The Visual C++ **Check out file(s)** dialog box appears, and the file that you selected is highlighted.

2. Click **OK** to check out the file.

Multiple files

To check out multiple files:

1. In the Visual C++ **File View**, select the files that you want to check out.
2. From the main menu, select **Project > Source Control > Check Out**.

The Visual C++ **Check out file(s)** dialog box appears, and the files that are pre-selected (highlighted) are all of the files you selected in step 1. You can select or de-select any file by clicking its check box.

3. Click **OK** to check out the selected files.

Checking out classes

Use the class **Check Out** option to check out a single, read-only class. From the main menu, select **Project > Source Control > Check Out** to check out multiple classes.

If you right-click a read-only class to open it and then initiate the check-out by modifying the text, you are checking out and modifying only the class method file. If you want to check out all files associated with the class, use the pop-up menu to perform the check-outs.

Note Ensure that you have selected the appropriate task **before** checking out any objects. Any objects you check out are associated with the selected task.

Single classes

To check out a single class:

1. In the Visual C++ **Class View**, right-click the class, and then select **Check Out**.

The Visual C++ **Check out file(s)** dialog box appears, and the files that are pre-selected (highlighted) are all of the files associated with the class you selected. You can select or de-select any file by clicking its check box.

2. Click **OK** to check out the selected files.

Multiple classes

To check out multiple classes:

1. In the Visual C++ **Class View**, select the classes that you want to check out.
2. From the Visual C++ main menu, select **Project > Source Control > Check Out**.

The Visual C++ **Check out file(s)** dialog box appears, and the files that are pre-selected (highlighted) are all of the files associated with the classes you selected. You can select or de-select any file by clicking its check box.

3. Click **OK** to check out the selected files.

Checking out resources

Unlike files and classes, resources are checked out when they are saved, rather than when they are first opened and modified. When you double-click a read-only resource folder, a message appears reminding you to check out a resource before modifying it. The message is for your information only and is not persistent.

Note Ensure that you have selected the appropriate task before checking out any objects. Any objects you check out are associated with the selected task.

Single resources

To check out a single resource:

1. Click the resource to view the pop-up menu.

If you check out the resource folder itself, all of the resources in the folder are checked out.

2. Select **Check Out**.

The Visual C++ **Check out file(s)** dialog box appears, and the files that are pre-selected (highlighted) are all of the files associated with the resource you selected. You can select or de-select any file by clicking its check box.

3. Click **OK** to check out the file.

Multiple resources

To check out multiple resources:

1. Click the resources that you want to check out.
2. From the Visual C++ main menu, select **Project > Source Control > Check Out**.

The Visual C++ **Check out file(s)** dialog box appears, and the files that are pre-selected (highlighted) are all of the files associated with the resources you selected. You can select or de-select any file by clicking its check box.

3. Click **OK** to check out the selected files.

Opening a file, class, or resource


When you double-click a checked-out source-code-controlled file, class, or resource from Visual C++, the corresponding file is opened, and you can make changes.

When you double-click a read-only source-code-controlled file, class, or resource and begin to modify it, you are prompted to check out the corresponding file.

If you click **Yes**, the **Check out file(s)** dialog box appears with any associated files highlighted that are not already checked out. See “Checking out files” on page 36, for the file check-out procedure.

Checking in files, classes, and resources

To check in a file, class, or resource, do one of the following:

- Select the object, and then click .
See “Completing tasks” on page 50.
- Select the object, and then from the Visual C++ main menu, select **Project > Source Control > Check In**.

If you are using task-based methodology, check in your selected task, which also checks in all files associated with that task.

Single files, classes, or resources

To check in a single file, class, or resource:

1. Right-click the object, and then select **Check In**.

The **Check in files(s)** dialog box appears. When you activate the **Keep checked out** option, a new file is checked out automatically for each file that you check in.

2. Click **OK**.

If you check in a resource folder instead of an individual resource, all of the checked-out resources in the folder are checked in.


Multiple files, classes, or resources

To check in multiple files, classes, or resources:

1. Select the objects that you want to check in.
2. From the Visual C++ main menu, select **Project > Source Control > Check In**.

The Visual C++ **Check in files(s)** dialog box appears, and the files that are pre-selected (highlighted) are all of the files associated with the objects you selected. You can select or de-select any file by clicking its check box.

When you activate the **Keep checked out** option, a new file is checked out automatically for each file that you check in.

3. Click **OK** to check in the files.
4. Once you have completed work on all files associated with a task, click  to complete the task.

See “Completing tasks” on page 50.

Changing the properties of Rational Synergy files

The database properties of a Rational Synergy file are a subset of the Rational Synergy properties of the file. Properties include the date the file was created, the date it was last modified, its status, and its owner. Rational Synergy automatically sets the Rational Synergy properties.

To change Rational Synergy properties:

1. In Visual C++, select the file for which you want to change one or more properties.

The file must be checked out in order to change properties. If it is not already checked out, do so now. Also, you may view the properties of a file that is not checked out.

2. From the main menu, select **Project > Source Control > Synergy Properties**.

The **Properties** dialog box appears.

3. Change the properties.

For more information about the Rational Synergy **Properties** dialog box, see the Rational Synergy Help.

4. Click **Apply** to apply the changes, and then click **Close**.

Showing file histories

When you make file-based changes, you can obtain predecessor and successor information for your file version by displaying the Rational Synergy **History View**.

To show a file history:


1. In Visual C++, click the file for which you want to view the history.
2. From the main menu, select **Project > Source Control > Show History**.

The Rational Synergy **History** dialog box appears.

Obtaining latest versions of files, classes, and resources

You can obtain the latest version of a file, class, or resource by using one of the following options:

- **Update**

Click  to update the file, class, or resource. This is the preferred method, because the `update` command evaluates each file in the project to determine which version of the file belongs in the project. For more information about using this option, see “Updating projects” on page 50.

- **Get Latest Version**

From the Visual C++ main menu, select **Project > Source Control**.

This selection should be used sparingly, as it may not produce the desired results. This option uses the Rational Synergy `use version` command. If you have not updated your project correctly, the `use version` command can result in an incomplete configuration. To obtain more information about the `use version` command before performing this operation, see the Rational Synergy Help.

Undoing check out

You can undo the check-out of one or more files, classes, or resources by doing the following:

1. In Visual C++, click the object(s) for which you want to undo the check-out.
2. From the main menu, select **Project > Source Control > Undo Check Out**.

The **Undo Check Out** dialog box appears, and the files that are pre-selected (highlighted) are all files associated with the objects you selected. You can select or de-select any file by clicking its check box.

This operation invokes the Rational Synergy `delete` command. With these options, the `delete` command removes the checked-out object and replaces it with the previous object that would have been chosen by the selection rules. If there are one or more parallel versions of your object, the selection rules may select a version other than the one you expected. Therefore, when you undo a check out, verify that the object version that replaces the undone version is correct.

For more information about the selection criteria, see the Rational Synergy Help for the `update` command.

3. Click **OK**.

Adding uncontrolled files, classes, or resources

Visual C++ allows you to add a new file, class, or resource to source code control when you create it. Rational Synergy transparently creates the database objects associated with the new object as part of the Visual C++ **Add to Source Control** operation.

If your environment requires the exporting of .mak files for build management, extra steps need to be taken when adding or removing files from your project because Visual C++ does not automatically check out a .mak file. For more information, see “Exporting makefile changes” on page 29.

Caution Do not use the **Add to Source Control** operation to load projects into Rational Synergy. Instead, use the **Migrate** operation. See “Migrating projects” on page 19.

To add an uncontrolled file, class, or resource:

1. In Visual C++, select your task.


See “Selecting tasks” on page 49.

2. Create and insert a new object into your project.

See “Adding files, classes, or resources to projects” on page 28.

If you are adding a product to your project, you must activate the **is_product** property. Use the Rational Synergy **Properties** dialog box to activate this setting.

See “Changing the properties of Rational Synergy files” on page 40.

3. Exit the object creation dialog box, and then make the necessary code changes.
4. Click  to complete the task.

This method automatically checks the .dsp file and directory objects into the Rational Synergy database, which does not happen when you individually check in the files.

See “Completing tasks” on page 50.

Deleting controlled files, classes, or resources

Delete a controlled file, class, or resource from your Visual C++ project to remove the files from your project in the database. The files are removed from the file system, but they are not deleted from your database.

After you have removed your file, class, or resource from the Rational Synergy database, you must delete it from your Visual C++ project.

If your environment requires the exporting of .mak files for build management, extra steps need to be taken when adding or removing files from your project because Visual C++ does not automatically check out a .mak file. For more information, see “Exporting makefile changes” on page 29.

To delete controlled files, classes, or resources:

1. In Visual C++, select the files, classes, or resources that you want to remove.
2. From the main menu, select **Project > Source Control > Remove from Source Control**.

The Visual C++ **Remove from Source Control** dialog box appears, and the files that are pre-selected (highlighted) are all of the files associated with the objects you selected. You can select or de-select any file by clicking its check box.

3. In **Comment**, type your comment, and then click **OK**.

The files are removed from the Rational Synergy database. However, the files are not removed from the Visual C++ workspace.

Closing projects

Whenever you close a project, for example, when you change from one project to another, the Rational Synergy **Close project** prompt appears.

If you have enabled the **Prompt to sync when closing project** option, you may be prompted to sync the work area for the project you are closing. If you have changed any files in your current project workspace and you want to ensure that the Rational Synergy database is updated with your changes, click **Yes**.

Enable this option in the **Synergy** advanced operations dialog box. See “Work Area dialog box” on page 47.

Displaying the Rational Synergy Work pane

There are many operations that extend the Visual C++ built-in capabilities that you can perform using the Rational Synergy for Microsoft Source Code Control Interface. These operations are often started from the Rational Synergy **Work** pane.

From the Visual C++ main menu, select **Project > Source Control > Synergy**.








The Rational Synergy **Work** pane should appear. If you do not see the **Work** pane, you might need to bring it to the front from the background.

Accessing task-based CM features

Rational Synergy enables you to create, select, and check in tasks using either special options on the **CMS Bar** toolbar or a tabbed dialog box, located in the **Synergy** advanced operations dialog box.

CMS Bar toolbar

On the **CMS Bar** toolbar, you can access the following task-based CM features:

- Select Task 
- Create Task 
- Complete Task 
- Show Task Properties 
- Sync 
- Update 
- Help 

Use to view the Rational Synergy for Microsoft Source Code Control Interface documentation.

To add the **CMS Bar** toolbar to the Visual C++ toolbar area, see “Customizing the Rational Synergy toolbar” on page 8.

Synergy advanced operations dialog box

The **Synergy** advanced operations dialog box includes three tabs: **Options**, **Work Area**, and **Task-Based CM**.

To access the **Synergy** advanced operations dialog box:

1. On the Visual C++ main menu, select **Tools > Options**.

The **Options** dialog box appears.

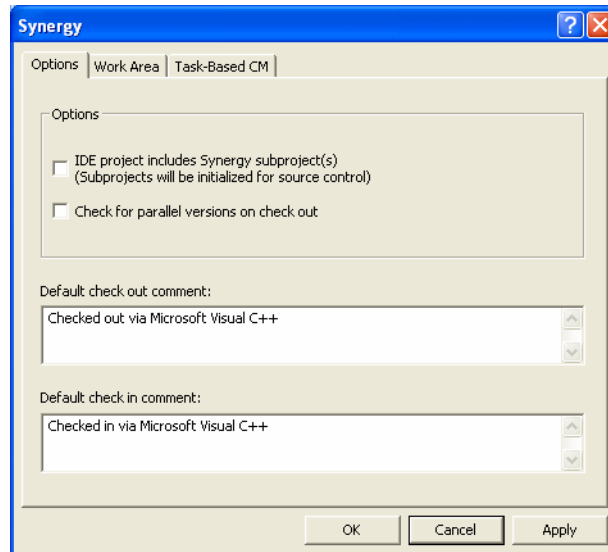
2. Click the **Source Control** tab, and then click **Advanced**.

The **Synergy** advanced operations dialog box appears.

Options dialog box

Use the **Options** dialog box to set the following options:

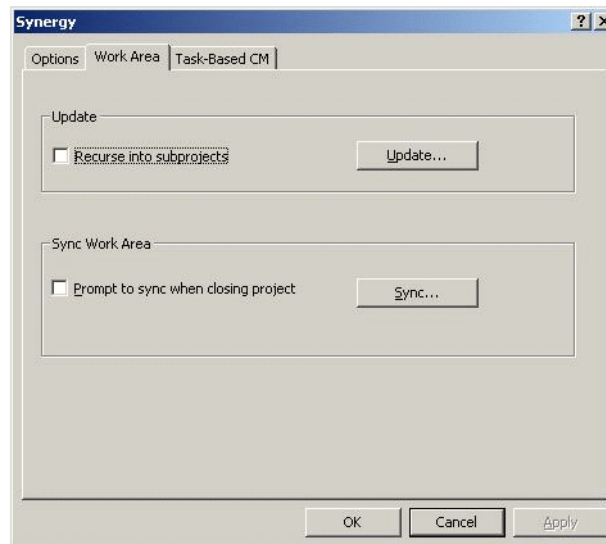
- **Subprojects for source control**—Allows you to perform source control on subprojects.
- **Check for parallel versions on check out**—Notifies you of parallel versions upon check out.
- **Default check out/check in comments**—Adds comments to the file properties upon check out and check in.



Work Area dialog box

Use the **Work Area** dialog box to set the following options:

- **Recurse into subprojects**—Updates the subprojects.
- **Prompt to sync when closing project**—Prompts user to sync the work area when closing projects.

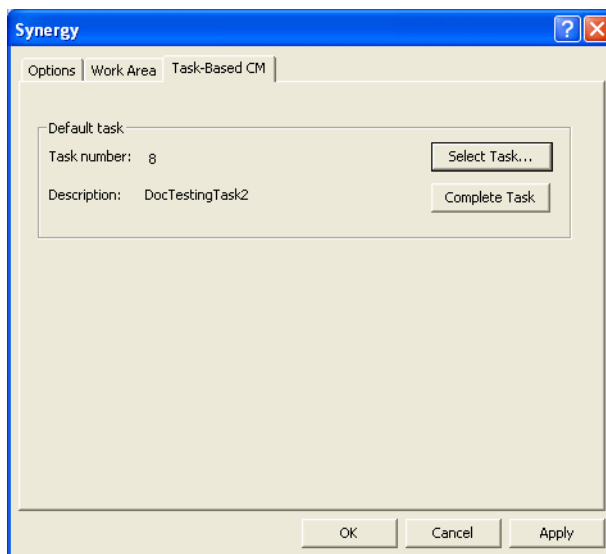


Task-Based CM dialog box

Use the **Task-Based CM** dialog box to do the following:

1. To select a task:
 - a. Click **Select Task**.
The **Select Task** dialog box appears.
 - b. In **Task ID**, click to select the task, and then click **OK**.
 - c. Click **Apply** to apply the changes.
2. To complete a task, click **Complete Task**.

When you complete the task, you are also checking in all files associated with the task. See also “Completing tasks” on page 50.



Selecting tasks

When you select a task, all CM operations that you perform during your session use that task number for files that do not already have one. For example, all files that you check out are automatically assigned to the selected task.

Note Ensure that you have selected the appropriate task **before** you check out any objects. You can select only one task per Rational Synergy session. If you are running multiple Visual C++ sessions, they share the same selected task. To avoid associating changes with the wrong task, run one Visual C++ session at a time.

To select a task:

1. Click .

The **Select Task** dialog box appears.

2. In **Task ID**, click to select the task, and then click **OK**.

You can also select a task from the **Task-Based CM** tab in the **Synergy** advanced operations dialog box. See “**Task-Based CM dialog box**” on page 48.

Creating tasks

If you are working on a new fix or enhancement that does not already have a task associated with it, you need to create a new task. This new task is assigned to you automatically, but you can change the **resolver** or any of the other task properties through the dialog box in which you are creating the task.

To create a task:

1. Click .

The **Create Task** dialog box appears.

2. Type a task synopsis that is a clear, concise description of the fix or enhancement (an entry is required).
3. Set any special properties for your task.

The choices contained in the drop-down list are either Rational Synergy defaults or customized values set by your CM administrator.


4. Click **OK**.

The task is automatically assigned to you, and the task fields are non-writable.

Completing tasks


When you complete a task, all files associated with that task are checked in. Then, the task is checked in.

To complete a task:

1. Click .
2. From the Visual C++ main menu, select **Project > Source Control > Refresh Status** to update your display.


You can also select a task from the **Task-Based CM** tab in the **Synergy** advanced operations dialog box. See “**Task-Based CM dialog box**” on page 48.

Updating projects

In Visual C++, when you click  to update a project using Rational Synergy, each file in the project is examined to determine which, if any, version of that file belongs in the project.

In task-based CM, only the files associated with the project tasks and its baseline are included. For more information about the update command, see the Rational Synergy Help.

Syncing projects

In Visual C++, when you click  to sync a project using Rational Synergy, changes in your work area are synced with changes in the database.

By default, if changes in the database are in conflict with changes in your work area, the work area changes are kept and the database changes are discarded. If you want to have more control over the conflict handling strategy, use the Rational Synergy **Sync** dialog box. For more information about the sync function, see the “Sync work area with database” section of the Rational Synergy Help.

Working in disconnect mode

The *disconnect* mode allows you to work in Visual C++ if you exit Rational Synergy or if Rational Synergy is not available. This mode is useful when you are working away from your usual workspace, such as working at a home office or using a disconnected laptop. In *disconnect* mode, all files are considered to be under source code control, permissions are preserved, and the only source code control operation available is **Check Out**.

Disconnecting

To disconnect:

1. In Visual C++, open the project.

The **Rational Synergy Startup Info** dialog box appears.

2. Click **Cancel**,

A message appears asking if you want to run in disconnect mode.

3. Click **Yes**.

The project is in disconnect mode.

Reconnecting

To reconnect:

1. Close Visual C++.
2. Reopen Visual C++, and then open your project.

The **Rational Synergy Startup Info** dialog box appears so you can log on to Rational Synergy.

When you start your next Rational Synergy session, you should sync your workspace from the Rational Synergy **Work** pane. The sync work area operation resolves any conflicts between your work area and your Rational Synergy database. For more information about the sync function, see the “Sync work area with database” section of the Rational Synergy Help.

Removing projects from source code control

Before removing a project from source code control, shut down all Rational Synergy sessions.

To remove a project from source code control:

1. In Visual C++, open the project.

The **Rational Synergy Startup Info** dialog box appears.

2. Click **Cancel**.

A message appears asking if you want to run in disconnect mode.

3. Click **No**.

A message appears asking if you want to reattempt this connection in future sessions.

4. Click **No**.

Clicking **Yes** disconnects the project and allows the connection to be reattempted in the future.

Subproject control

The integrated development environment (**IDE**) project option includes Rational Synergy subprojects (subprojects will be initialized for source control), which allow you to initialize (recurse) your project and its subprojects. Activate this option if your IDE application includes Rational Synergy subprojects and you want to perform source control operations on members of your subprojects.

If your project contains subprojects but you do not need to perform source control operations on them, do not set this option. The default (option is disabled) setting initializes only the top level project on startup.

Startup times will be faster if only the top level application is being initialized.

- Turn on this setting if you want to perform source control operations on subprojects.
- If your application does not contain any subprojects, this option can be ignored.

Parallel version control

When you check out a file, you can set the system to automatically check for parallel versions.

Default check out and check in comments

When you check out or check in a file, you can add a comment to the comment property of file. Thereafter, when you look at the Rational Synergy properties of the file version, you see the comment.

The default comment for files that you check out or in using Visual C++ is “Checked in/out via Microsoft Visual Developer Studio.”

Appendix: 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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
5 Technology Park Drive
Westford, MA 01886
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.

Trademarks

IBM, the IBM logo, ibm.com, and Rational Synergy are trademarks or registered trademarks of International Business Machines Corporation 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.

Microsoft, Windows, Windows 2003, Windows XP, Windows Microsoft IDE Integration Types, Visual C++, Visual Basic, Visual Studio, and/or other Microsoft products referenced herein are either trademarks or registered trademarks of Microsoft Corporation.

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

Index

A

- absolute paths, 13
- adding
 - new projects, 27
 - objects, 27
 - uncontrolled objects, 43
- administrator privileges, installing interface, 9
- advanced operations
 - Options dialog box, 46
 - Task-Based CM dialog box, 48
 - Work Area dialog box, 47
- automated builds, setting up, 23

B

- baselines, creating, 24
- build management projects, copying, 31
- build process, factors affecting, 32
- build strategy, 16

C

- ccm_scc.dll file, 3
- checked-out objects, editing, 27
- checking in
 - objects, 39
 - single and multiple objects, 39
- checking out projects, undoing, 42
- classes, checking out, 37
- CMS Bar toolbar, 45
- command line, building from, 16
- commands
 - update, 13, 41, 42, 50
- controlled objects
 - deleting, 44
 - source code control options, 3
- controlling uncontrolled objects, 43
- creating tasks, 49
- current data, viewing, 26

D

- default comments, 53
- deleting controlled objects, 44
- disconnect mode, 51
- DLL file, 3

E

- editing checked-out objects, 27

F

- factors affecting build process, 32
- files
 - changing Rational Synergy properties, 40
 - checking out, 36
 - determining which version to use, 50
 - obtaining latest versions, 41
 - showing history, 41
 - working outside the work area, 51

G

- getting started, 21

H

- history, file, 41

I

- IBM Customer Support, 4
- interface
 - how it works, 1
 - installing, 7
- is_product property, 43

M

- makefiles, managing, 16
- MAPI, 3
- Microsoft file types, installing, 9
- migration
 - migrating projects, 19
 - verifying, 22

multiple classes, checking out, 37
multiple resources, checking out, 38

N

new projects, adding to workspace, 27

O

objects
 editing checked-out, 27
 opening, 39
 removing, 28
obtaining latest versions of files, 41
Options dialog box, 46

P

products, setting is_product property, 43
projects
 adding new, 27
 baselines, creating, 24
 closing, 44
 configuration changes, 27
 copying, 31, 36
 migrating, 19
 reconciling, 44, 50
 setting up, 21
 update, 50
 updating with latest information, 32
properties, changing, 40

R

Rational Synergy
 accessing task-based features, 45
 bring objects under control, 43
 project path information, 13
 properties, changing, 40
 subprojects, 13
 Work pane, 45
Rational Synergy for Microsoft Source
 Code Control Interface
 how it works, 3
 objects it controls, 2

reconciling projects, 44, 50
refreshing your screen, 26
relative paths, 13
removing objects, 28
resources, checking out, 38

S

saving changes, 24
SCCI, 3
selected tasks, number allowed per session,
 26
single classes, checking out, 37
single objects, checking out, 36, 37
single resources, checking out, 38
source code control options, visual cue, 3
subprojects
 changing location, 14
 compiling and linking movable, 14
 paths, 13
 performing source control on, 52
 Rational Synergy, 13
 updating locations, 13
 Visual C++, 12

T

Task-Based CM dialog box, 48
task-based methodology, using, 25
tasks
 completing, 50
 creating, 49
 selecting, 49
timestamp
 option, 17, 32, 33
 setting option, 23
toolbar, customizing, 8

U

undoing check out, 42

update
 changes, 27
 command, 13, 41, 42, 50
 operation, 13
 prior to builds, 32
 projects, 50
use operation, 13

V

Visual C++
 building from command line, 33
 dependency information, 12
 dependent subprojects, 14
 modifying settings, 15
 reference subprojects, 14
 running multiple sessions, 26
 subprojects, 12

W

Work Area dialog box, 47
Work pane, displaying, Rational Synergy,
 45

