

RIDE

Part 1: Programming



Reference manual

Revision 01/2000

RAISONANCE

The information included in this manual may be modified by RAISONANCE S.A and is in no way a commitment of RAISONANCE S.A. The rights of use of the software described in this document, may only be transferred under a licensed agreement. Thus, the software may only be used under the terms of this agreement. All copying of this document for any other purpose besides the personal use of the buyer, must have prior authorization from RAISONANCE SA. The present manual covers the RIDE software.

Copyright 2000 RAISONANCE SA. All rights reserved.

REGISTERED TRADEMARKS:

Raisonance is a registered trademark of RAISONANCE S.A.

Contents

1. GENERAL PRESENTATION	7
1.1 Organization	8
1.2 RIDE for which tools?	9
1.3 Presentation of the documentation	10
2. INSTALLATION AND INVOCATION	11
2.1 Installing RKit	12
2.2 System requirements	12
2.3 Installation procedure	13
2.3.1 To run RKit installation program	13
2.3.2 Setup type	13
2.3.3 Directory	13
2.3.4 User information	13
2.3.5 To uninstall RKit	13
2.4 Invocation of RIDE	15
3. RIDE INTERFACE	17
3.1 Interface Basics	18
3.2 The text editor	19
3.3 The Menu Commands	20
3.4 The Tool Bar	20
3.5 The Status Bar	21
3.6 Dockable windows	22
4. PROJECT MANAGER	23
4.1 What is a project?	24
4.1.1 The project window	24
4.2 Project Menu	26
4.2.1 Creating a new project	26
4.2.2 Opening an existing project	28
4.2.3 Adding source files to the project	28
4.3 Tool menu	29
4.3.1 Library manager	29

4.3.2	Running a tool	29
4.4	Listing views	29
4.4.1	Map file	29
4.4.2	LST file	29
4.4.3	Messages window	30
5.	SETTING OPTIONS WITHIN A PROJECT	31

5.1	Target Options	32
5.1.1	Derivative selection	32
5.1.2	Case sensitivity	32
5.2	Compilation Options	32
5.2.1	Source options	33
5.2.2	Code generation options	33
5.2.3	Defines options	34
5.2.4	Listing options	34
5.2.5	Object options	35
5.2.6	Memory Model options	35
5.2.7	Optimizer options	36
5.2.8	Messages options	36
5.3	Assembler Options	37
5.3.1	Source options	37
5.3.2	Set options	37
5.3.3	Listing options	38
5.3.4	Object options	39
5.4	Linker Options	40
5.4.1	Linker options	40
5.4.2	More controls	40
5.4.3	Listing	40
5.4.4	Bank switching	42
5.5	Environment Options	43
5.5.1	Directories	43
5.5.2	Colors	44
5.5.3	Font	44
5.5.4	Editor	45
5.5.5	Execution	46
5.6	Tools Options	47
5.7	Debugging Options	50
5.7.1	Configuration	50
5.7.2	Running another tool	50
6.	COMPILATION	51

6.1	Compiling a file	52
6.2	Making a project	52

6.3 Building a project	53
6.4 Linking a project	53
7. EDITING	55
7.1 Editor Options	56
7.2 Manipulating Files	56
7.2.1 File filters	56
7.2.2 Customizing the file filters	56
7.2.3 Creating a new source file	56
7.2.4 Opening a source file	57
7.2.5 Saving a file	58
7.2.6 Printing files	58
7.2.7 Exiting from RIDE	58
7.3 Editing a Text	59
7.3.1 Fast moving in the source file »	59
7.3.2 Using OEM characters »	59
7.3.3 Using Tabulation characters	59
7.3.4 To transfer text	59
7.3.5 To copy text	59
7.3.6 To undo an editing action	60
7.4 Finding and replacing text	60
7.4.1 To find a text string	60
7.4.2 To replace a text string	62
7.5 Window management	63
7.6 « Help »	64
7.7 commands summary »	65
8. INDEX	67

1. General presentation

1.1 Organization

1.2 RIDE for which tools ?

1.3 Presentation of the documentation

RIDE is an Integrated Development Environment (« IDE ») containing everything you need to write, edit, compile, link and debug your microcontrollers' applications:

- Editor,
- Translation tools: Macro-Assemblers, C Compilers,
- Utility tools: Linker, Library Manager...
- Real Time Kernel,
- Debug tools: Simulator or Emulator.

1.1 Organization

This documentation introduces you to application development with **RIDE**. It consists of two parts corresponding to the main steps of the development mechanism.

- The first part of the documentation (« Part 1: Programming ») provides an overview of the editor and the different integrated utility and programming tools.
- The second part (« Part 2: Debugging ») describes the interface of the debugging tools: Simulator or Emulator. This second part is delivered only to the users of these debugging tools.

The programming tools have their own documentation, so this document discusses only about their interfaces.

RIDE is available in a 32-bit version (designed for Windows 95, 98 and NT).

1.2 RIDE for which tools?

The Windows versions of RKit compiler, assembler and simulator are Dynamic Link Library files « .DLL » which could be used from **RIDE**.

RIDE could be configured to work with a changing number of tools. In general, the initial configuration is set by the installation process and the serial number of the user will determine the tools that could be fully used and the ones that are usable as evaluation versions.

For the ST6 family two kits are proposed:

- AKIT-ST6 kit containing:
 - the MA-ST6 Assembler
 - the RL-ST6 Linker
 - the SIM-ST6 Simulator/Debugger
- RKIT-ST6 kit containing:
 - the RC-ST6 Compiler
 - the MA-ST6 Assembler
 - the RL-ST6 Linker
 - the SIM-ST6 Simulator/Debugger

1.3 Presentation of the documentation

This documentation consists of 8 chapters:

1. **Introduction**
2. **Installation and invocation**
Contains information about installing, configuring, and running **RIDE**.
3. **Presentation of the interface**
Provides an overview of **RIDE** user interface. It describes the basic components of the environment, such as the toolbar, status bar, and Help system.
4. **Project Manager**
Describes how to create and use projects to build your applications.
5. **Project options**
How to set the projects' environment and configure compiler, assembler and linker options.
6. **Compilation:**
RIDE lets you run programming and debugging tools of your choice without leaving the interface environment. This chapter explains how to install tools and run them from the main menu.
7. **Editing**
Discusses how to use the editor to create, open, save, and close files, edit text, move around in files, search for text, and other editing tasks.
8. **Index**

2. Installation and invocation

2.1 Installing RKit

2.2 System requirements

2.3 Installation procedure

2.4 Invocation of RIDE

2.1 Installing RKit

An autorun.inf file makes the installation start automatically and you just have to follow the instructions. If it is not the case, the « RKit-Installation » program (INSTALL.EXE) installs all the necessary files and correctly configures the environment for running **RIDE**. You can install the kit from CD-ROM or a network. The files on the distribution disks are compressed; the installation program both uncompresses the files and copies them to your hard disk. Before installing **RKit**, make sure that your computer meets minimum system and language requirements and close any other running application. The « README.TXT » file, from the first setup disk, contains important information about the installation procedure.

2.2 System requirements

RKit requires the following minimum configuration:

- A 80486 processor (or better) running one of the following operating systems:

Operating System	Memory Required
Microsoft Windows 95, 98 or NT	8 megabyte (MB) of RAM

- A hard-disk drive with 22 megabytes of free space. (The actual space required depends on the options you select. Additional space is needed for your source-code files and the executable files created from them.)

2.3 Installation procedure

2.3.1 To run RKit installation program

- 1 Insert the CD into your CD-ROM drive.
- 2 The installation starts automatically. If not, from the Windows Explorer or File Manager, select « x:\INSTALL.EXE » (x is the letter representing your CD-ROM drive), and press ENTER.
- 3 Follow the instructions on the screen.

2.3.2 Setup type

The installation program will install on your computer:

RIDE interface

Select this button to install the **SIMICE** simulator. This is installed in the \BIN directory. This option installs both DOS and Windows versions of **SIMICE**.

MA-ST6 Macro assembler

The assembler and its utilities are installed in the \BIN directory. The include files are installed in the \INCLUDE directory. The examples are installed in the \EXAMPLES\ST6\MAST6 directory.

SIM-ST6 Simulator

This is installed in the \BIN directory

2.3.3 Directory

By default, the root directory for the installation is set to « C:\Ride ». If you want to change this installation directory, type the complete path in the **Destination Location** dialog box.

2.3.4 User information

You will find your Serial Number on your CD or on the registration card attached to the documentation. This serial number is necessary for the installation procedure and must be registered properly. This number will define the tools that will run as evaluation versions and the ones that will be completely available. Don't lose this number; you must use it every time you install the software.

2.3.5 To uninstall RKit

1. In the Start Menu, choose **Program\RKit**
2. Click on Uninstall. The RKit directory and its files will be deleted.

If your Operating system is WINDOWS 95

3. Open Regedit.
4. Open the Local Machine Folder

5. Open the Software folder.
6. Open the Microsoft folder.
7. Select the RAISONANCE folder.
8. From the Edit menu, choose Delete. The entries for RIDE are deleted from the Windows registry.

2.4 Invocation of RIDE

After you have installed **RKit**, start **RIDE** by double-clicking its icon in the Program **Manager** window.

Ride-IDE: This program gives you access to the Program Manager where you can create, build and debug new applications. The full command is RIDE.EXE.

Ride-Reference: The On-line Help for RIDE. The full command is RIDE32.HLP.

If you had a project open when you last exited **RIDE**, it is automatically loaded. To quit **RIDE**, choose Exit from the File menu. This returns you to the Windows Program Manager.

3. RIDE interface

3.1 Interface Basics

3.2 The text editor

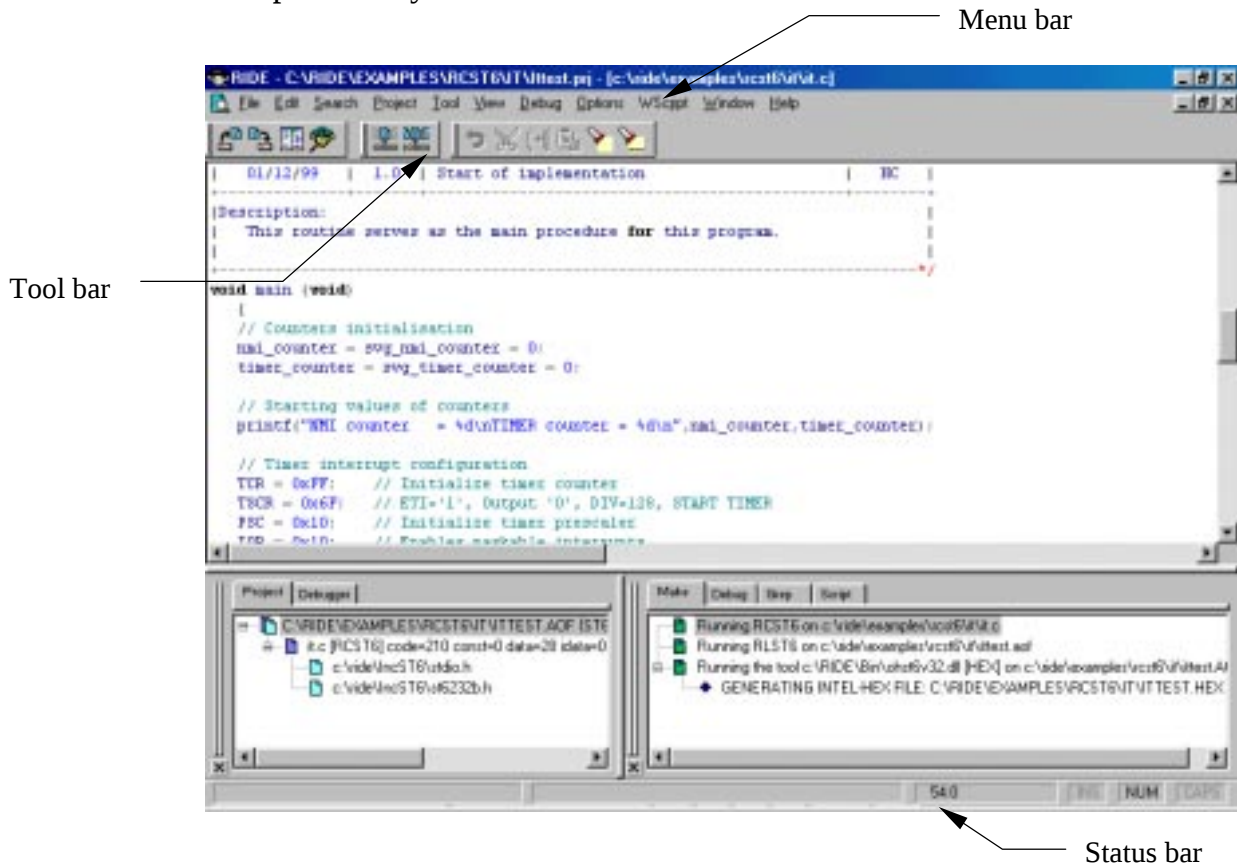
3.3 The menu commands

3.4 The tool bar

3.5 The status bar

3.1 Interface Basics

RIDE is an Integrated Development Environment consisting of a text editor, project manager, programming tools' manager, debugger, and Help database. This chapter briefly introduces **RIDE**.



RIDE interface

3.2 The text editor

RIDE provides an integrated syntax highlighting the text editor to manage, edit, and print source files. Most of the editor's procedures like file and text handling, and moving around in a file, are the same as in other Windows-based text editors.

With the editor, you can:

- Perform advanced find and replace operations.
- Perform bracket matching.
- Specify syntax coloring.
- Customize tab's length in source files.
- Use toolbar shortcuts for various commands.
- Get context-sensitive Help from within a source file.
- Use multiple levels of undo.
- Open multiple Windows for debugging, displaying source files, and viewing project structure.

3.3 The Menu Commands

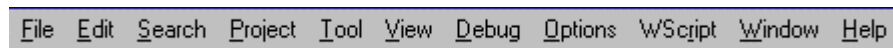
The menu bar is the primary access to the commands. Each menu consists of a list of items that can be viewed or chosen. When the action represented by a command is not appropriate, the menu item is grayed text and does not respond to mouse clicks or keyboard selection.

To display a menu using the keyboard, you can:

- Press Alt then press the hot key in the menu pad name. The hot key is underlined and is usually the first letter in the menu pad name.
- Press F10 then press the hot key in the menu pad name.

If you are currently in the menu bar, a menu pad is already highlighted. You can display a menu by pressing the hot key in the menu pad name. You can also press the Left and Right Arrow keys until the menu pad is selected, then press the Down Arrow key to display the menu.

To get help about a menu command, you can highlight the command and then press F1 to display the corresponding help topic.



RIDE Menu Bar

3.4 The Tool Bar

The toolbar is a row of bitmapped buttons, placed beneath the menu bar that activates commands. Pressing a toolbar button is similar to choosing a menu item.

Buttons on the toolbar are grouped according to function.

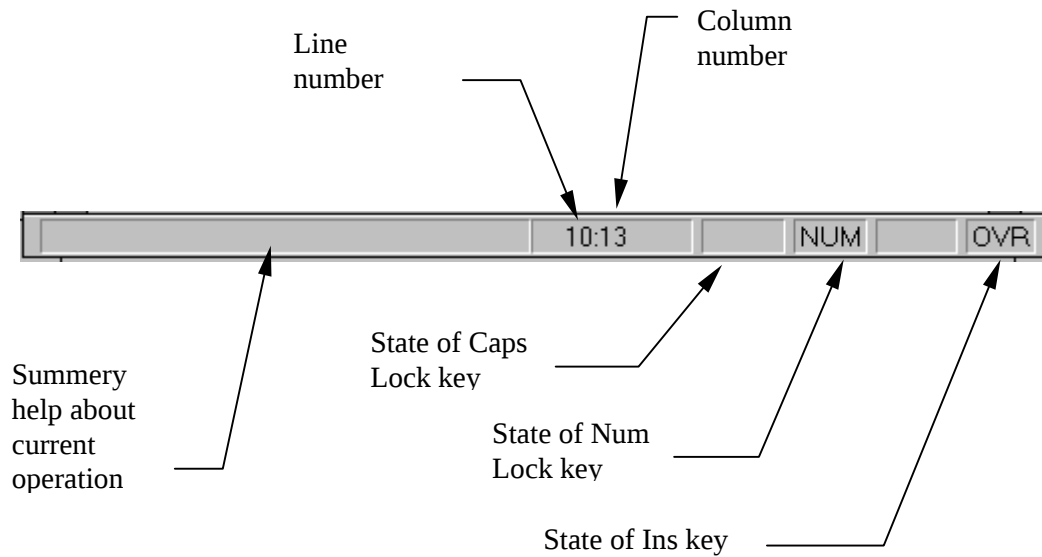


RIDE Tool Bar

3.5 The Status Bar

The status bar is a control bar with a row of text, or "status indicators". The status bar appears at the bottom of the main window of RIDE.

The status bar shows the following information about the active window:



RIDE Status Bar

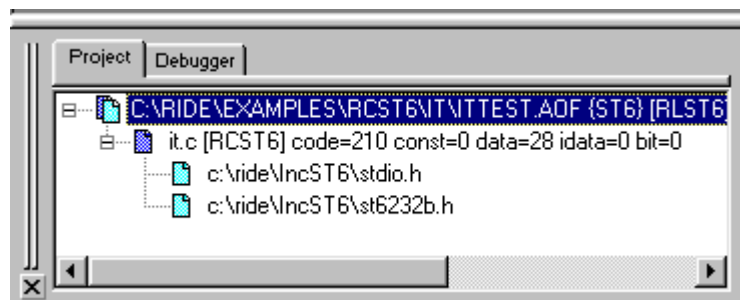
3.6 Dockable windows

Ride presents the Project and the Message windows as dockable windows.

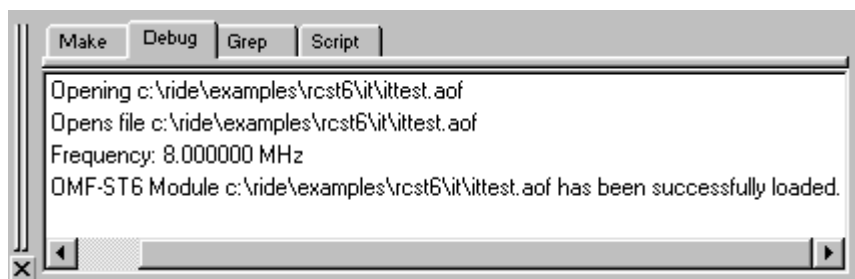
Dockable windows have two modes: floating or docked. In floating mode, a window has a thin title bar and can be moved anywhere on your screen. A floating window is always on top of all other windows. In docked mode, a window is fixed to a dock along any of the four borders of the main Ride window.

To show a docking window, you have to click the name of the window on the View menu. To hide the window, you have to click the Hide button in the corner of the window. Not that you can show or hide the Project, Messages windows at any time. Hiding the project window does not mean closing the project workspace. Note that the window positions for docking windows are not associated with the current project; they remain the same no matter which project you open.

To position a floating window over a dock, you have to point to the title bar of the window and drag the window over any dock area of Ride application window. To change a docked window to a floating window, have to point the frame of the window and drag it out of the docking area.



Project Dockable window



Messages Dockable window

4. Project manager

4.1 *What is a project ?*

4.2 *Project menu*

4.3 *Tools menu*

4.4 *Listing views*

4.1 What is a project?

The Project Manager helps you to manipulate and work with complex applications. A project contains information about a particular program, its files and its environment, its programming tools and their options. These information are registered in two files: a file containing the project options (.PRJ extension) and a file containing the debug options « .WMC » .

RIDE retains the last project you were working with during your last session. It will display the same desktop of this project for the new session. The project file contains three kinds of information:

- The names and locations of the source files and their dependencies that are used to build your program.
- The settings for the tools required building the program, such as compiler, assembler, linker or kernel options.
- The organization of the RIDE desktop on your display.

4.1.1 The project window

The project window is a dockable window. Tabs at the top of the project window provide different ways to view your project. The project views are “Project” and “Debugger”.

The “Debugger” view is set during a debug session and gives a quick access to the memory and hardware windows.

The “Project” view shows a graphical representation of the project. It displays the different files, their paths and associated tools, which constitute the program/application. You can set options, add or delete files from a project only if this window is displayed. The project has a hierarchical structure that may be navigated in a manner similar to Windows Explorer.

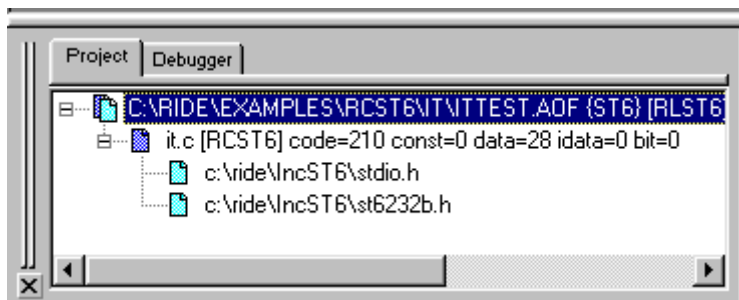


Figure 1 RIDE project's view

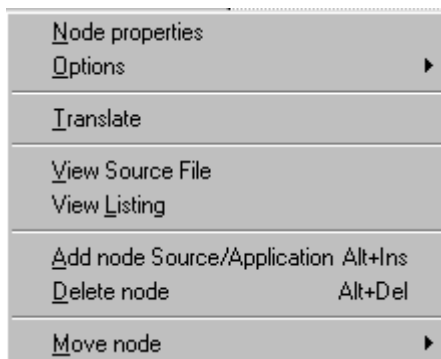
Within RIDE, the project's application is represented as the root of a tree. Each file belonging to it is represented as a node on the tree. Indented below each node are the node's dependencies used to build the node.

Next to the application name, the type of the project (ST6) and the result of the last successful link (code Size, internal data Size and external data Size) are

listed. Next to the application nodes, the build translator, and the result of the last successful compilation or assembly are listed.

The project options are inherited by the project nodes. You can override these options by setting locally options on specific nodes. When a new node is added to the project, according to its file extension, a build translator is automatically associated with it. You can select another build translator using the pop-up menu commands.

If you press the right mouse button within this window, the project's pop-up menu will appear.



- **Node properties:** This command displays a window with the different information concerning the current node (location, translating tool, last modification...).
- **Options: Local Options:** This command permits the user to set locally the build options on the currently selected node. If local options are used to build a node, in the project view, it is marked with .
- **Options: Global Options:** This command lets the user re-select the global project options to build the currently selected node.
- **Translate:** This command builds the selected node with its associated build translator.
- **View Source File:** This command displays the content of the selected node. You can also view the content of a node with selecting the node and pressing ENTER.
- **View Listing:** This command displays the content of the listing file generated by the last compilation or assembly of the node.
- **Add node Source/Application:** With this command you can insert a new node in the project's list. The file filters proposed for selecting files in the **Open File** dialog box correspond to the programming tools associated with the project.
- **Delete node:** This command suppresses the selected node reference from the project's list. The command suppresses the file from the project's node but not from the disk. If you want RIDE to prompt you the name of the file before suppressing it, check the corresponding option in **Options|Project|Environment|Editor**.
- **Move Node: Up:** This command moves up the selected node in the tree. You can also use Shift+UpArrow to do so.
- **Move Node: Down:** This command moves down the selected node. You can also use Shift+DownArrow to do so.

4.2 Project Menu

The main menu contains a sub-menu entitled « Project », which provides access to 3 groups of commands:

- New, Open, and Close
- Add File and delete File
- Compile, Make, Build all and Link.

A pop-up menu is also available when the project windows is open. To display this menu, select a node (file) of the project and press the mouse right button. This menu allows the insertion of a new file, the suppression of a selected file, or its display in an editing window. The modification of the project options can also be carried out from this menu.

4.2.1 Creating a new project

The first step to develop a program with RIDE is to create a project. To do that:

1. From the **Project** menu, choose **New**. The **New Project** dialog box appears.
2. Click the **Browse** button to display the **Open New Project** dialog box.
3. Choose the directory and the name of your project file.

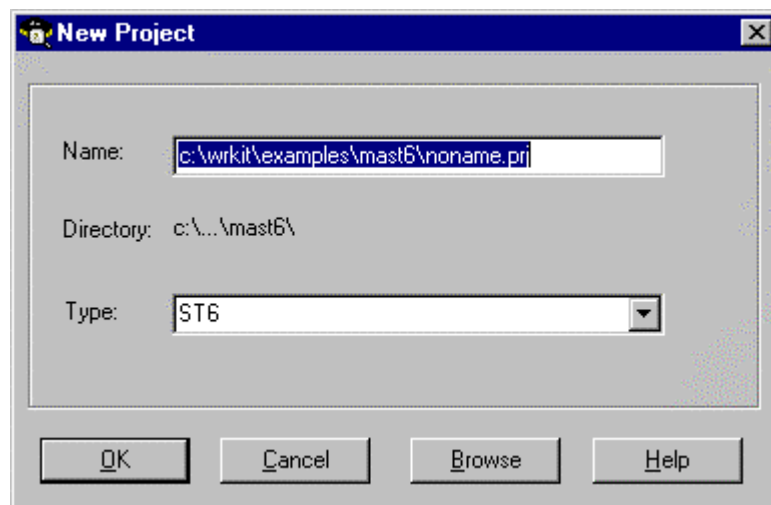


Figure 2 RIDE New project dialog box

4. Choose **OK**. The name and the directory you have chosen will be displayed in the **New Project** dialog box.

You have to select the main options of the program:

- **Model**: this option lets you specify your target model.
- **Libraries**: this options lets you specify the libraries you want to work with.

The choice of the project type will select the main characteristics of the project. It will set the default building tools and configure the project's environment according to these selected tools.

If this project exists already, RIDE prompts you to know if you want to overwrite the old project file. When the project is created, RIDE change the working directory to the project directory and displays the Project window. So you can add source files to your project and set its different options.

4.2.2 Opening an existing project

To open an already existing project:

- Choose **Open** from the **Project** menu. The **Open** dialog box appears.
- Choose the directory and the name of the project you want to open and choose **OK**.

RIDE will display the project window and the working desktop of this project. You will find the same opened windows and options setting than your last working session.

4.2.3 Adding source files to the project

You can use menu commands to add and remove files from your project. You can also use the popup-menu of the project window.

4.2.3.1 To add files to a project

1. From the **Project** menu, choose **Add file**. The **Add File** dialog box appears.
2. You can also choose **Add file** in the pop-up menu opened with clicking the right mouse button in the project window.
3. Select the file type to display from the **List Files of Type** drop-down list box. The files of this type appear in the **File Name** list box. The extensions listed in this box could be changed with changing the default C source files and assembler files in the projects options. Several files could be selected at the same time.
4. If necessary, select the drive and directory to view.
5. Add files with the following methods: Select the file in the **File Name** list and choose **OK**. Alternatively, double-click the file in the **File Name** list.

4.2.3.2 To remove a file from a project

1. Select the file in the Project window list.
2. From the **Project** menu, choose **Delete file**. Alternatively, from the pop-up menu displayed with clicking on the right mouse button, choose **Delete file**.

The default tool associated to a source file depends on its extension. Files with the same extension, as the C source files will be processed with the C compiler. In addition, files with the same extension as the assembler files will be processed with the assembler. If the file doesn't have a standard extension, you have to associate to the file a tool. You can use the pop-up menu of the project window to select a build tool for the node.

4.3 Tool menu

4.3.1 Library manager

The Library Manager dialog box, lets you modify your libraries without leaving the RIDE. The commands are:

- **Path:** Type the path of your library directory in this edit control. The list of the different library files will be displayed in the Library Files list box.
- **Library Files:** This list box lets you select the library file you want to modify.
- **Modules:** This list box displays the different modules defined in the currently selected library in the Library Files list box.

Click the bitmap  close to the name of the module to display the list of the Publics Symbols or  to hide them.

- **Add button:** Press the Add button to add a new module to the library currently selected in the Library Files list box.. Pressing this button displays a dialog box to let you choose the name and location of the new object file to insert in the library.
- **Delete button:** Press the Delete button to remove the module currently selected in the Modules list box from the library.

4.3.2 Running a tool

With the **Run tool** command, you can run a DOS application without leaving the RIDE environment. The command proposes a list of tools where you can select the one you want to execute. To add or delete an application from this list use the **Options|Tools** command.

4.4 Listing views

4.4.1 Map file

This command displays the listing file (**.MS6**) corresponding to the project if it exists.

RL-ST6 generates listing files that contain the following information about the link process:

1. The invocation command line.
2. Object modules included in the project.
3. The memory map.
4. The overlay map.
5. The symbol table of public, local, and line number information.
6. A cross reference table.

4.4.2 LST file

Only with RIDE, this command displays the listing file (".**LST**") corresponding to the listing file generated for the currently selected node from the project or opened window.

4.4.3 Messages window

This command makes the messages window of the current project active.

5. Setting options within a project

5.1 *Target Options*

5.2 *Compilation Options*

5.3 *Assembler Options*

5.4 *Linker Options*

5.5 *Environment Options*

5.6 *Tools Options*

5.7 *Debugging Options*

To set options for your project. You can choose the kind of options from the **Options** menu. You can also use the global or local option setting with the pop-up menu which appears if you press the mouse right button in the Project window. This Menu lets you set local options for processing the selected source file or choose the global options of the project for this processing.

5.1 Target Options

RIDE offers a global level of settings, common for all the tools.

5.1.1 Derivative selection

In the case of the ST62 family, the main choice concerns the selection of the derivative reference.

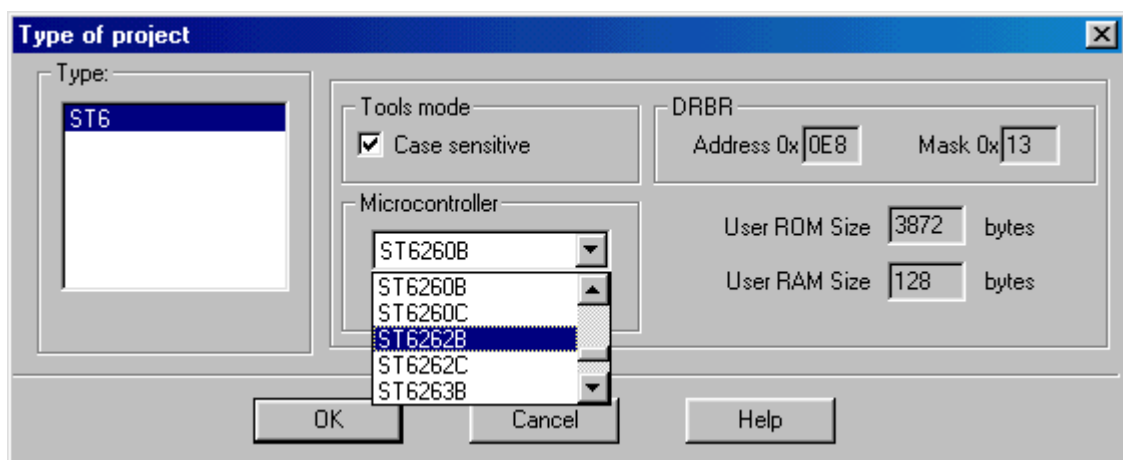


Figure 3 Target options in RIDE

IMPORTANT

The different tools will use this information to set properly the memory configuration. This information will be used by most of the tools:

- The programming tools will consider different addresses for DRBR, and different masks depending on the selected derivative (the meaning and the address of DRBR is variable),
- The debugging tool will use this information to define automatically the list of the internal peripherals and also the memory size.

5.1.2 Case sensitivity

Raisonance tools are NOT case sensitive by default. It means that, for the MA-ST6 assembler, the symbol “var1” will be considered as identical to “VAR1” or “Var1”. Nevertheless, the C compiler, during the process time, will assume “case sensitiveness” in any case (as defined by ANSI specification). Conclusion:

- If “Case Sensitive” is selected, the rule will be applied to all the tools.
- If it is NOT selected, the rule will be applied to the assembler and linker.
- The ST-AST6 assembler is always case sensitive, when it is an option for MA-ST6.

5.2 Compilation Options

From RIDE, to set the compilation options, choose the **Project** command from the **Options** menu, and select RCST6. You can set the following options:

5.2.1 Source options

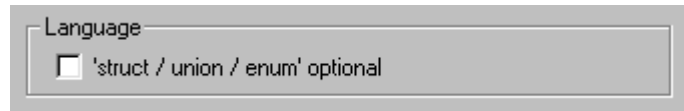


Figure 4 Compilation Source options in RIDE

- **'struct / union / enum' optional:** For a variable declaration of a struct, union or enum type, the keyword struct, union or enum can be omitted. By default this option is not checked.

5.2.2 Code generation options

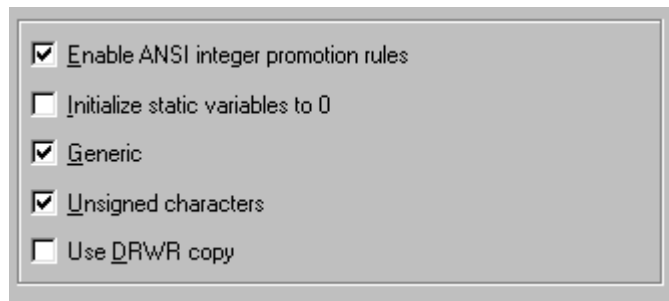


Figure 5 Compilation Code generation options in RIDE

- **Enable ANSI integer promotion rules:** The ANSI C standard requires promoting characters to integers. With an 8-bit processor, this conversion (or promotion) could generate ineffective code. This option lets the user choose to enable or disable this promotion rule. This option is checked by default.
- **Initialize static variables to 0:** The ANSI standard imposes that global variable and static variable must be initialized to 0 when there is no specific initialization value. Its is possible to disable this option to make the initialization faster. This option is unchecked by default.
- **Generic:** If this option is selected any pointer without a space qualifier will be used as a generic pointer. If this option is not selected any pointer without a space qualifier will be considered as pointing to the default space (XDATA for the LARGE and HUGE models, and DATA for the SMALL and TINY models). It is recommended to keep always the same mode. It is easier to work with the Generic mode but it is less efficient. This option is checked, by default.
- **Unsigned characters:** If this option is selected the 'char' types are unsigned by default. Checked by default.
- **Use DRWR copy:** If this option is selected a copy of DRWR is used for IT/NMI. By default this option is unchecked.

5.2.3 Defines options

Preprocessor defines used in #if statements and macro definitions to the preprocessor can be entered on the Compiler Defines page. The following rules apply when using the Defines input box: Separate multiple defines with semicolons (;) and assign values with an equal sign (=) or a backslash(/).

5.2.4 Listing options

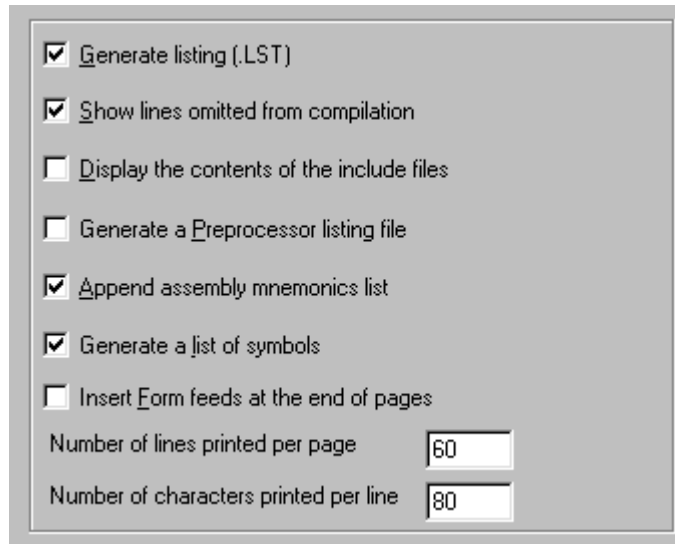


Figure 6 Compilation Listing options in RIDE

- **Generate Listing(.LST):** This option enables or disables the compiler generating a listing file. By default, the listing file is generated.
 - **Show lines omitted from compilation:** RC51 supports conditional compilation by replacing the appropriate source-code lines with a blank line. If this option is checked, the lines of code, which are not compiled in a conditional structure, will be included in the listing file. By default, this option is checked
 - **Display the contents of the include files:** If this options is checked the listing file will include listing information generated by the compilation of files included using the #include command. By default, this option is not checked.
 - **Append assembly mnemonics list:** If this option is selected, the listing file will contain the generated code in assembly language. Checked by default
 - **Generate a preprocessor Listing File:** If this option is checked, the listing file will be generated after preprocessing. By default, this option is not checked.
 - **Generate a list of symbols:** If this option is checked, the listing file will include the list of symbols defined in the application.
 - **Insert form feeds at the end of pages:** If this option is checked, form feeds are included in the listing file in order to permit a proper printing of the file.
 - **Number of lines printed per pages:** Default is 60
 - **Number of characters printed per line:** Default is 80
- These two input box let you define the format of the listing file.

5.2.5 Object options

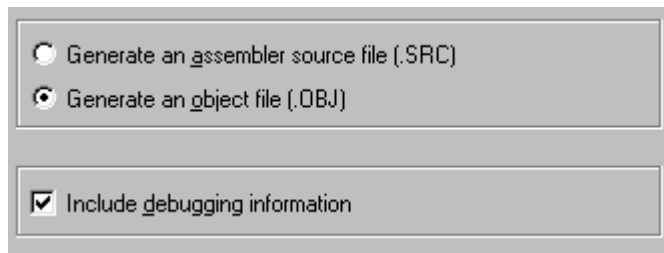


Figure 7 Compilation Object options in RIDE

- **Generate an assembler source file(.SRC):** If this option is checked, the compiler generates an assembler source file (.SRC) in spite of an object file. This file could be assembled by the assembler MAST6.
- **Generate an object source file(.OBJ):** If this option is checked the compiler generates an object file and no assembler source.
- **Include debugging information:** These options enable or disable the generation of symbol tables and line numbers in the resultant .OBJ file.

5.2.6 Memory Model options



Figure 8 Compilation Memory Model options in RIDE

Small / Large (Memory): These check boxes lets you select the memory model to be implemented. The available options are **Small**, and **Large**. The compilation model defines the default memory type to be used for local variables and other declarations without explicit memory type, as well as the stack location..

5.2.7 Optimizer options

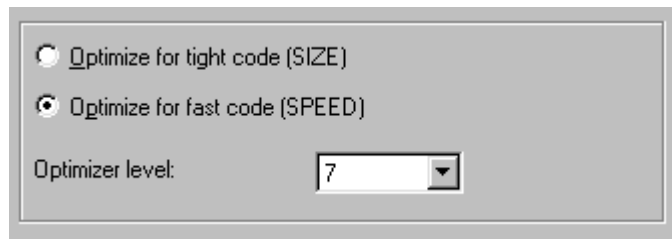


Figure 9 Compilation Optimizer options in RIDE

- **Optimize for tight code (SIZE):** If this option is checked, the compiled code is optimized for size.
- **Optimize for fast code (SPEED):** If this option is checked the compiled code is optimized for speed.
- **Optimizer level:** There are seven levels of optimization. Please refer to the compiler documentation for more details.

5.2.8 Messages options

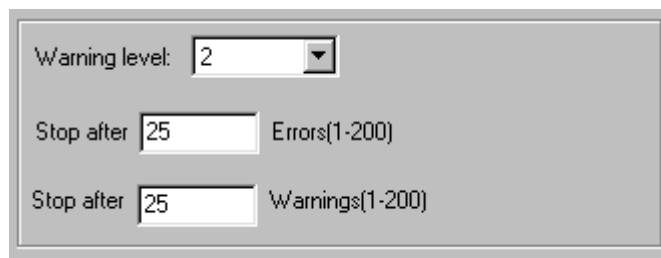


Figure 10 Compilation Messages options in RIDE

- **Warning level:** Depending on the selected warning level this option enables or disables the compiler to generate warning messages; this option doesn't affect the error messages.
- **Stop after *number* Errors (1-200)** This option makes the compiler stop after the specified number of errors found in the source file.
- **Stop after *number* Warnings (1-200)** This option makes the compiler stop after the specified number of warnings found in the source file.

5.3 Assembler Options

From RIDE, to set the assembler options, choose the **Project** command from the **Options** menu, and select MAST6. You can set the following options:

5.3.1 Source options

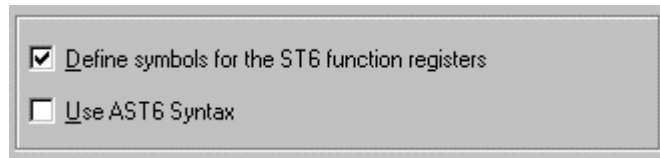


Figure 11 Source options in RIDE

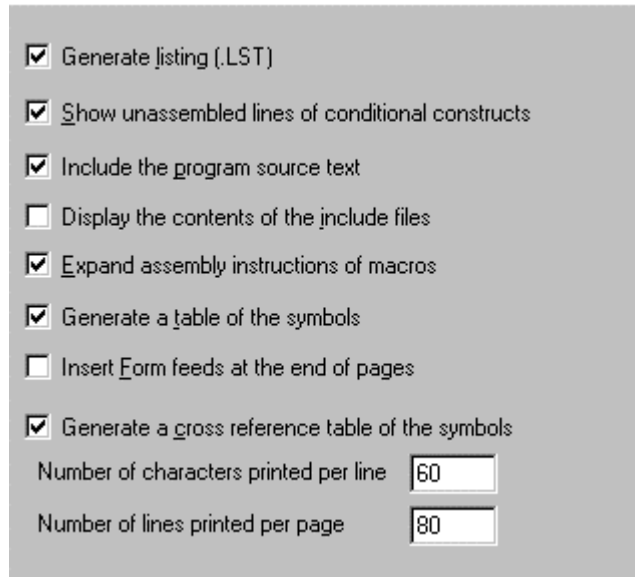
- **Define symbols for the ST6 ...:** This option permits the assembler to define symbols for the ST6 special function registers. This option is not checked by default.

- **Use AST6 Syntax** This option permits the assembler to take into account the syntax of the AST6 assembler. This option is not checked by default.

5.3.2 Set options

- **Set:** The Set input box permits the assigning of a numerical value or register symbol to a specified symbol name. Separate multiple sets with commas (,) and assign values with an equal sign (=).

5.3.3 Listing options



The screenshot shows a dialog box with the following options and settings:

- ☒ Generate listing (.LST)
- ☒ Show unassembled lines of conditional constructs
- ☒ Include the program source text
- ☐ Display the contents of the include files
- ☒ Expand assembly instructions of macros
- ☒ Generate a table of the symbols
- ☐ Insert Form feeds at the end of pages
- ☒ Generate a cross reference table of the symbols
- Number of characters printed per line:
- Number of lines printed per page:

Figure 12 Set options in RIDE

- **Generate Listing(.LST):** If this option is checked, the assembler generates a listing file(.LST). By default, this option is checked.
- **Show unassembled lines of conditional constructions:** MAST6 supports conditional assembling by replacing the appropriate source-code lines with a blank line. If this option is checked, the lines of source, which are not assembled in a conditional structure, will be included in the listing file. By default, this option is checked.
- **Include the program source text:** This option enables or disables the assembler to include the program source text in the generated listing file. By default, this option is checked.
- **Display the contents of the include files:** This option enables or disables the assembler to include the source of the include files. By default, this option is not selected.
- **Expand assembly instructions of macros:** If this option is not checked the MAST6 assembler prevents from including macro expansion text in the listing file. Only the macro name is listed. By default, this option is checked.
- **Generate a table of symbols:** This option directs the assembler to include in the listing file a table of all symbols used in the assembly program module. By default, this table is generated.

- **Generate a cross-reference table of the symbols:** This option directs the assembler to include the cross-reference table of the used symbols in the listing file. By default, this table is not generated.
- **Insert form feeds at the end of pages:** This option inserts form feeds at the end of pages in order to make a proper print of the listing.

The following input boxes let you define the format of listing pages.

Number of characters printed per line

Number of lines per printed page

5.3.4 Object options

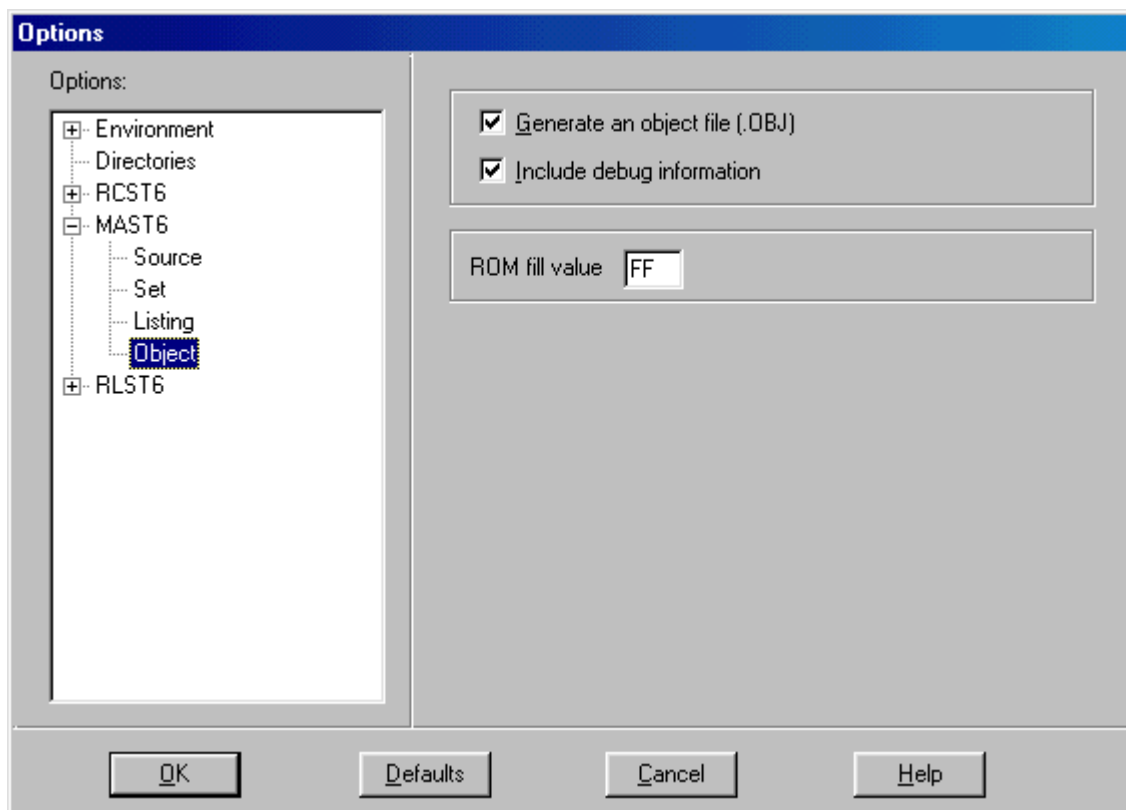


Figure 13 Object options in RIDE

- **Generate an object file:** This option directs the assembler to generate an object file (.obj).
- **Include debug information:** This option includes debug information in the object file generated by the assembler.

ROM fill value

Enter the initialization value of the ROM reserved bytes.

5.4 Linker Options

From RIDE, to set the ST6 linker, choose the **Project** command from the **Options** menu, and select RLST6. You can set the following options:

5.4.1 Linker options

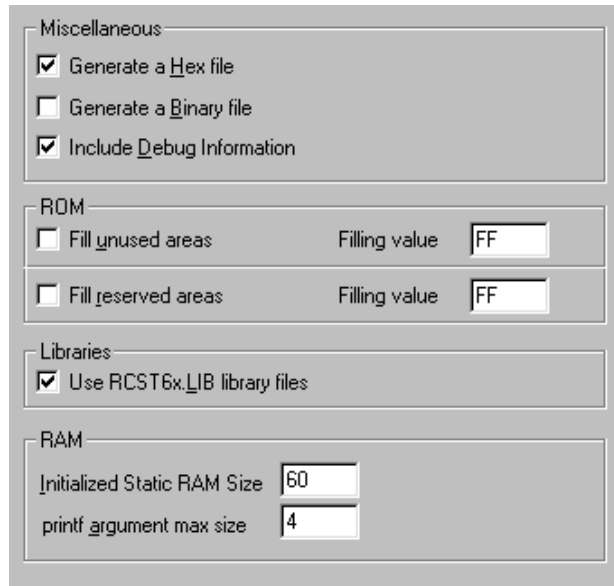


Figure 14 RL-ST6 options in RIDE

■ Misc

1. **Generate a Hex file:** Generate an Hex output file. Checked by default.
2. **Generate a Binary file:** Generates a Binary file within the project. Not checked by default.
3. **Include Debug Information:** Debug information can be kept in the output file. Checked by default.

■ ROM

1. **Fill Unused areas:** Select this option to fill the unused areas of the ROM. By default this option is not checked. **Filling value:** Value to put in the unused areas of the ROM.
2. **Fill Reserved Areas:** Select this option to fill the reserved areas of the ROM. By default this option is not checked. **Filling value:** Value to put in the reserved areas.

■ Libraries

Use RCST6x.LIB library files: Select this option to use the RCST6x.LIB library.

■ RAM

1. **Initialized Static RAM Size:** Value from 0 up to 64.
2. **printf argument max size:** Number of bytes necessary in the stack segment of printf for the argument passing.

5.4.2 More controls

This control allows adding specific commands, like pragmas.

5.4.3 Listing

The listing options specify the format of the linker's listing file pages.

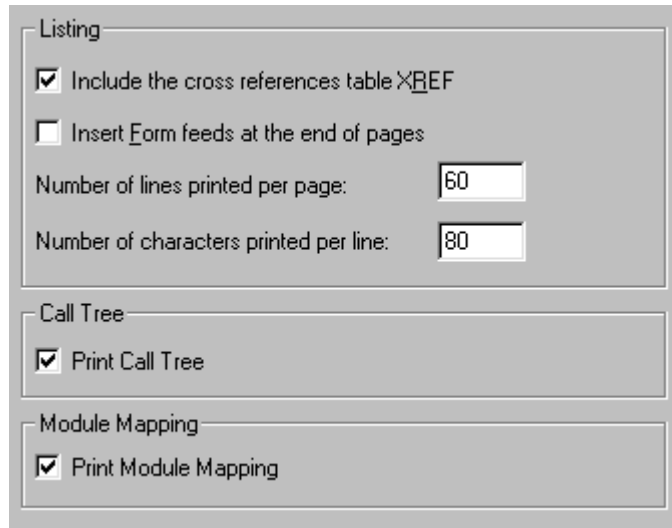
The image shows a screenshot of the 'Listing' options dialog box in the RIDE software. The dialog is titled 'Listing' and contains three main sections. The first section, 'Listing', has a checked checkbox for 'Include the cross references table XREF', an unchecked checkbox for 'Insert Form feeds at the end of pages', and two input fields: 'Number of lines printed per page:' with the value '60' and 'Number of characters printed per line:' with the value '80'. The second section, 'Call Tree', has a checked checkbox for 'Print Call Tree'. The third section, 'Module Mapping', has a checked checkbox for 'Print Module Mapping'.

Figure 15 RL-ST6 Listing options in RIDE

■ Listing

1. **Include the cross reference table XREF:** Generates a cross-reference file (.XRF).
2. **Insert Form feeds at the end of pages:** Useful, when the listing file is printed.
3. **Number of lines .. / Number of characters ..:** Configures the printing output.

■ Call Tree

Print Call Tree: Useful, when the listing file is printed.

■ Module Mapping

Print Module Mapping.

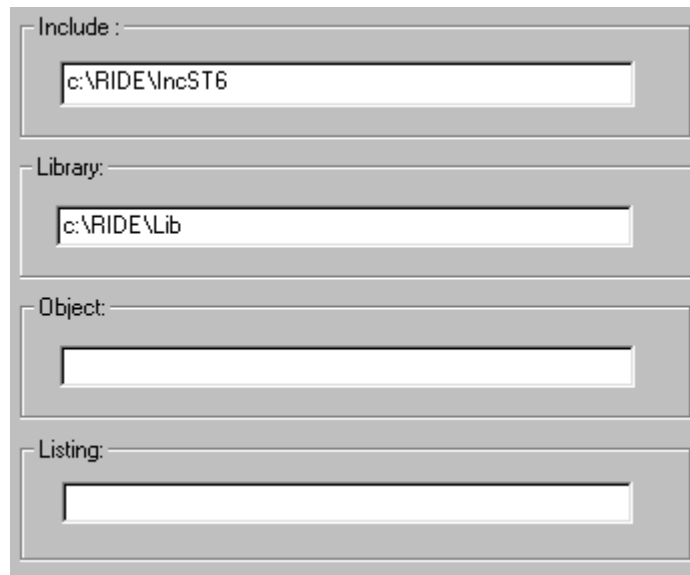
5.4.4 Bank switching

This command displays the Bank switching menu, where you can set the Bank switching Options. Refer to the RL-ST6 reference manual.

5.5 Environment Options

From RIDE, to set the Environment options, choose the **Project** command from the **Options** menu, and select Environment or Directories. You can set the following options:

5.5.1 Directories



The image shows a dialog box titled 'Directories options' with four input fields. The 'Include:' field contains 'c:\RIDE\IncST6'. The 'Library:' field contains 'c:\RIDE\Lib'. The 'Object:' and 'Listing:' fields are empty.

Figure 16 Directories options in RIDE

- **Include:** This input box specifies the directories that contain the standard include files. Multiple directory names are allowed in each of the input boxes; a semicolon (;) must separate the names.
- **Library:** This input box specifies the directories that contain the library files. Multiple directory names are allowed in each of the input boxes; a semicolon (;) must separate the names.
- **Object:** This input box specifies the directory where RIDE places the object (.OBJ) files and the generated (.AOF) file.
- **Listing:** This input box specifies the directory where RIDE places the listing (.LST and .MS6) files.

5.5.2 Colors

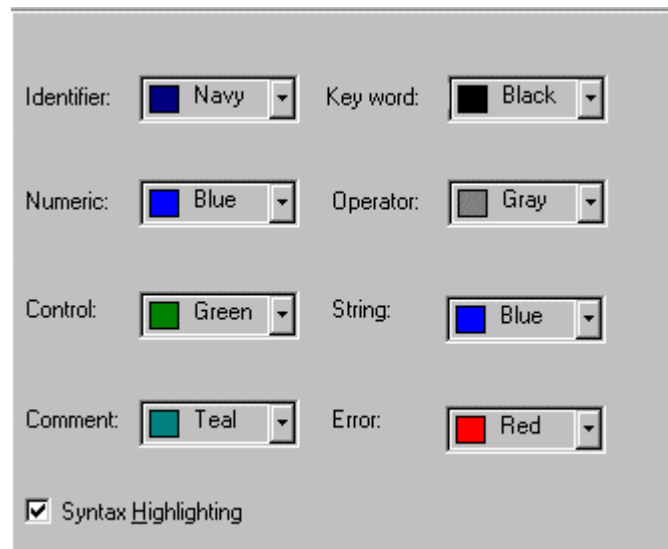


Figure 17 Colors options in RIDE

See 5.5.1.4

5.5.3 Font

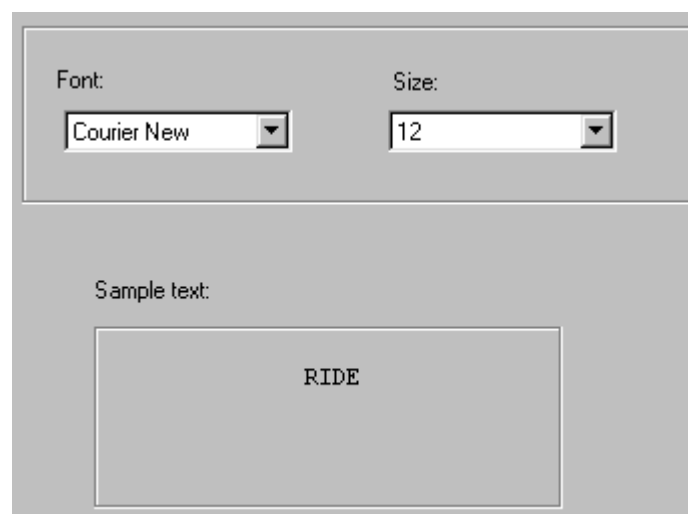
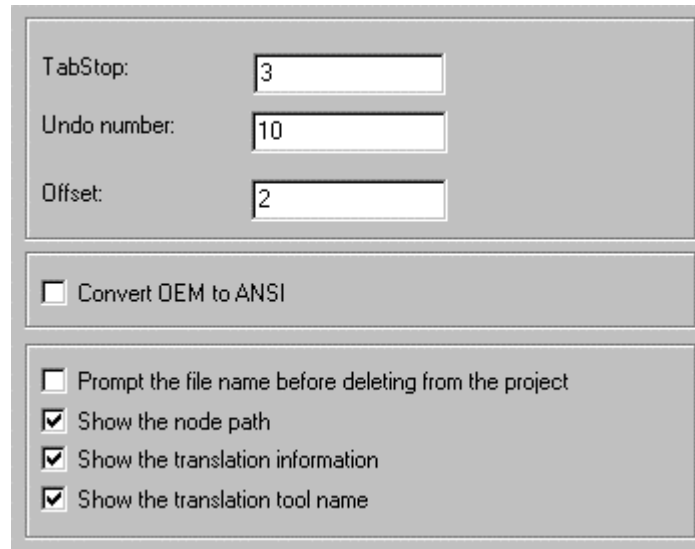


Figure 18 Font options in RIDE

See 5.5.1.5

5.5.4 Editor



TabStop:

Undo number:

Offset:

☐ Convert OEM to ANSI

☐ Prompt the file name before deleting from the project

☒ Show the node path

☒ Show the translation information

☒ Show the translation tool name

Figure 19 Editor options in RIDE

- **Tab Stop:** You can use tab characters in your source files. This control lets you set the number of spaces a tab consists of.
- **Undo number:** This input box lets you specify the number of editing actions available for being undone by the Undo command.
- **Offset:** This input box allows the user to specify the Right Margin for edit windows.
- **Convert OEM to ANSI:** If this option is checked, characters are read from files as OEM characters and saved as OEM characters. When editing the file they are converted to ANSI in order to make them compatible to the Window display.
- **Prompt the filename before deleting from the project:** If this option is checked, RIDE displays a Message Box asking for confirmation before deleting nodes in the project window.

5.5.5 Execution

The execution options let you enable or disable RIDE to prompt Message Boxes to signal the following events.

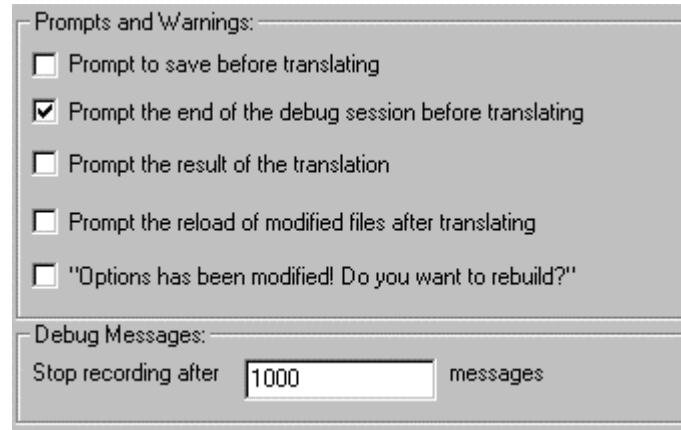


Figure 20 Execution options in RIDE

- **Prompt to save before translation:** If an editing window is opened and the text modified, RIDE will open a message box , and will ask you for saving or not the modification, before a Compile, Make or Build ALL command. By default: no message and auto-saving.
- **Prompt the end of the debug session before translating:** RIDE will open a message box , and will ask you to leave or not the debugging session, before a Compile, Make or Build ALL command. By default, the message box is opened.
- **Prompt the result of the translation:** RIDE will open a message box and will announce you the result of the translation. By default, no message.
- **Prompt the reload of the modified files after translating:** During the translation of a source file, this file is modified, RIDE will open a message box , and will ask for re-loading the modified file from the disk and translate this modified file. By default, no message and auto-reloading.

5.6 Tools Options

The **Options|Tools** command opens the **Tools** dialog box, where you can install, delete or modify the list of the project's tools:

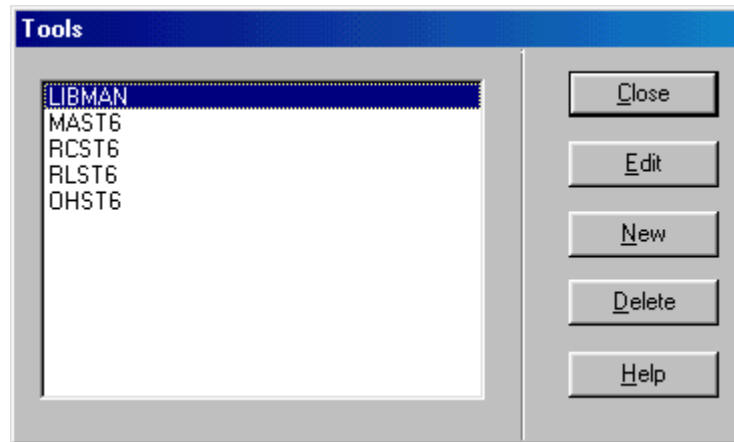


Figure 21 Tools options in RIDE

A user defined tool can be a debugging tool, a utility or a translator.

From this dialog box you can:

- **Edit:** This command allows to define or modify the tool.
- **New:** This command allows to create a new tool.
- **Delete:** select the name of the tool you want to suppress and click the **Delete** button.

The Tool Options dialog box lets you enter or edit the properties for the selected tool:

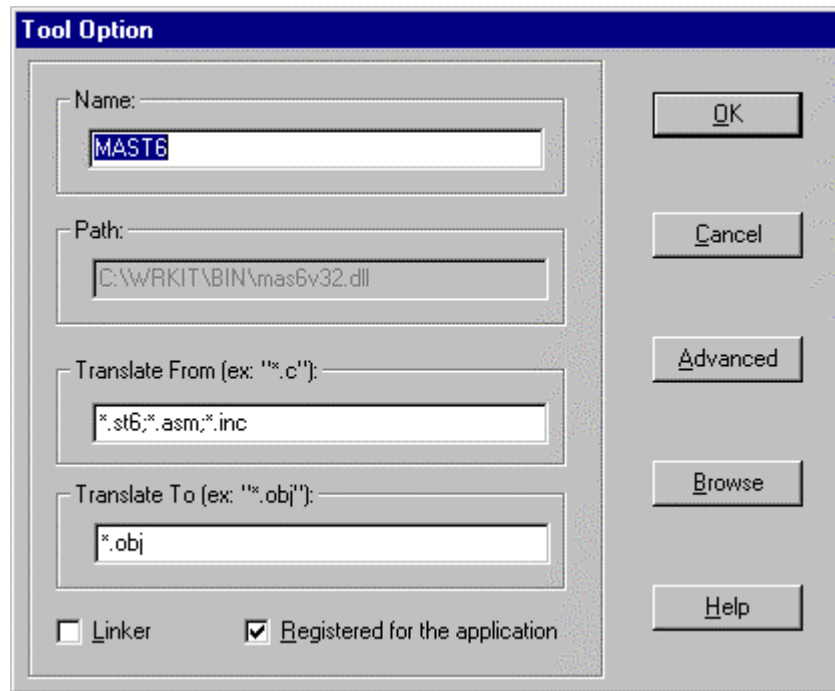


Figure 22 Tool options dialog box in RIDE

- **Name:** The Name input box lets you define a name that will designate the tool in the project environment. This is name will appear in tools lists. Remember that a project for ST6 will consider the tool named RCST6 as the default C compiler, the tool named MAST6 as the default assembler and the tool named RLST6 as its linker.
- **Path:** Type the complete path of the considered program. You can also click the Browse button to search for the path and filename of your program.
- **Translate From:** This input box lets you define the types of nodes that can use the tool. You can enter more than one file extension by separating them with semicolons (;). RIDE will use these extensions to select automatically the translating tool for each file. RIDE will also use these extensions to select the appropriate syntax highlighting for the file.
- **Translate To:** Use the Translate To input box to enter the file extension for the type of files that results from applying the tool to the file types defined in the Translate From input box.
- **Linker:** This option lets RIDE know that if the tool is a linker or not. Select this option if you want to use the tool as a linker.

- **Registered for the application:** With this option, you can add the defined tool to the list of RIDE default tools. Which means that for each new project, the tool will be automatically added to the project's list of tools. If this option is not selected the tool will remain specific to the active project. Selecting this option has no effect for the existing projects. RIDE doesn't modify the old projects but will look for modifications when creating new projects.
- **Advanced Options:** This button opens the Program options dialog box and lets you define the way that RIDE has to run the tool. If RIDE is not able to find the tool, it will not give access to these advanced options.

5.7 Debugging Options

5.7.1 Configuration

To set the Debug options, choose **Debug** from the **Options** menu. The **Debug Options** dialog box appears where you can choose the different Debug options:

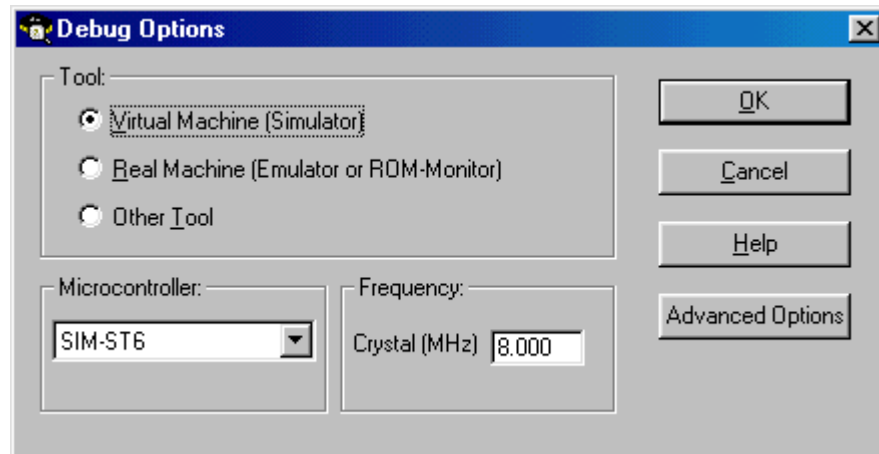


Figure 23 Debugging options in RIDE

- To use the integrated debugger, choose **Virtual Machine (SIMICE)**.
- To use the In-Circuit Emulator, choose **Real Machine (Emulator)**.
- To use another debugging tool, choose **Other tool**.
- To configure the microcontroller choose it the **Options|Target** menu.
- Enter the clock frequency in the **Crystal (MHz)** window (only for **Virtual Machine** mode).
- Configure the Port settings (only for **Real Machine** mode).
- Open the **Advanced Options** box and set the application options.

See **RIDE Part 2** documentation for more information about the advanced options and the debugging process.

5.7.2 Running another tool

RIDE allows you to run any other DOS debug program without leaving its environment. To do so, you have to:

1. Define a new tool in the project's environment. Insert the name of the application in the tools arguments in order to run it directly on the project's application.
2. Then set this tool as the project's debugger: From the **Options|Debug** menu command, select **Other tool** radio button and push the **Advanced Options** button. The **Select a Debugging Tool** dialog box will be displayed where you can choose your defined tool.

The command **Debug|Start** will automatically run this program without leaving RIDE.

6. Compilation

6.1 *Compiling a file*

6.2 *Making a project*

6.3 *Building a project*

6.4 *Linking a project*

From RIDE, you can build or rebuild the program that a project defines. When you choose to make a project, RIDE processes only the files in the project that have changed since the last build. When you choose build all command, RIDE processes all the files in the project.

When you create a project, RIDE sets default options for the target. By changing the compiler and the assembler options, you modify the compilation process.

6.1 Compiling a file

The **Project|Translate** command applies to a single file.

To compile a single file, you can activate this command or select the corresponding node in the project view, then use the popup menu of the project window. RIDE will process the file with the tool associated to the file.

The compilers and assemblers generate two files:

1. A listing file, which contains information relating to the compilation, including a list of any errors detected.
2. The object file containing the generated code in OMF-ST6 format.

These files have the same name as the source file, but with extensions “ .OBJ ” for the object files and “ .LST ” for the listing files. The output directory is set from the **Project|Options** dialog box.

A Compile Status information box displays the compilation progress and results. If any errors occurred, the Message Window becomes active, displays and highlights the first error or warning.

Message Window

Warning, error messages generated during the compilation and the link ,are displayed in the Message window.

Within RIDE, the error messages are marked with a red bitmap and the warning messages with a blue one.

The Message window provides an automatic error tracking. As you track up and down through the messages in the Message window, the cursor in the activated window moves to the corresponding source file line and highlights it, where you can fix the error. When a message refers to a file that is not currently loaded in a window, select the error and then press **Enter** to load the file. RIDE open the corresponding file and select the line that caused the error. The **Previous Message** and **Next Message** commands from the **Search** menu, help move from one error to another without selecting the corresponding message in the message window.

You can also get more information about an error by selecting the error in the Message window and pressing F1.

6.2 Making a project

To make a project choose the command **Make** from the **Project** menu.

The project manager will **MAKE** the project target. It checks file dates and times to see if they have been updated since the last build. If so, the project manager rebuilds those files, then checks the next node's file dates and times. The project manager checks all the nodes in a project and also their dependent files and builds all of the out-of-date files.

The Make command rebuilds only the files that are not current.

To decide whether a source file must be compiled or not, the Make utility refers to the dates of the object file and of the source file(s). The applied rules are as follows:

- If the object file does not exist, the source is compiled.
- If the source file is later than the object file, the source file is compiled.
- If the source file refers to « include files » which are later than the object file, the source file is compiled (automatic control of the dependencies).
- Otherwise, the source is not compiled

These rules apply to both C source files and assembly source files.

6.3 Building a project

The **Build all** command rebuilds all the files in the current project, regardless of whether they are out of date.

This option is similar to **Project|Make**, except that **Build All** rebuilds all the files in the project.

Note: The compilation is performed from the hard disk, and not from the memory of the edited file. Hence, RIDE prompts you to save the file before running the compilation. However, it is possible to make it automatically (see Options-Compiler).

6.4 Linking a project

The Link command links the project without building the nodes.

7. Editing

7.1 *Editor options*

7.2 *Manipulating files*

7.3 *Editing a text*

7.4 *Finding and replacing text*

7.5 *Window management*

7.6 *Help*

7.7 *Commands summary*

7.1 Editor Options

The editor options are saved within the project. Please refer to the paragraph 5.1.8 and 5.1.9 to set these options.

7.2 Manipulating Files

7.2.1 File filters

RIDE provides the handy feature of letting you use your own special filename filters in the **Open File** and **Save As** dialog boxes. These dialog boxes recall the filename filter that was last used. So you can use this feature to specify special file types that you always want to see.

The different available file filters are:

C Files: correspond to the filename extensions used for C and C header files. By default they corresponds to « *.C » and « *.H ».

Assembler Files: correspond to the filename extensions used for assembler and assembler header files. By default they correspond to « *.ST6» for the ST6 assembler files and to « *.pdf » for the include files.

Listing Files: correspond to the filename extensions used for the listing files of the compiler « *.LST », the assembler « *.LST » and the linker « *.MS6 ».

All Files: correspond to the filename extensions used for any kind of files(*.*).

The filename extension will determine the kind of syntax highlighting to be used for editing the file and also the programming tool to associate automatically with the file.

7.2.2 Customizing the file filters

RIDE lets the user customize its environment by modifying the preceding file filters.

The default file filters are associated with their corresponding tools. For example, to modify the C compiler file filter:

1. From the **Options** menu, choose **Tools**. The **Tools** dialog box appears.
2. Select from the list box the name of the project's C compiler.
3. Choose **Edit**, to open the **Tool Option** dialog box.
4. Type a new file filter in the **Translate From** control.
5. Choose **OK**.

7.2.3 Creating a new source file

To create a new source file:

1. from the **File** menu choose **New** (Ctrl+N). The **Document Type** pop-up menu appears.
2. Select your file's type from this menu. RIDE displays an **untitled** window. The editor will color the window's text according to the file type.
3. From the File menu, choose **Save As**. The **Save As** dialog box appears.
4. In the **Directories** box, double-click a directory where you want to store the source file (or move down a path to the appropriate directory).
5. Type a filename in the **File Name** box, then choose OK. The default extension given to a file is the last extension used when you saved a file. You can type another extension or select one from the **List Files Of Type** box.

The maximum number of bytes in a line is 255.

Before you can save or close an individual window it must be active. To make a window active, either switch to the window (by clicking anywhere in it) or choose the window name or number from the Window menu.

7.2.4 Opening a source file

To open a source file:

1. Choose **Open** (Ctrl+O) from the **File** menu. The **Open** dialog box appears.
2. Select the drive and directory where your file is. The default is your current drive and directory.
3. Set the types of files to display in the **List Files Of Type** box. Files with the chosen extension are displayed in the **File Name** box. The default extensions for C and assembler files could be modified with the Environment Options.
4. In the **File Name** box, select a filename, then choose **OK**. Or double click the filename.

The **List Files Of Type** list box serves as a filter in displaying files to open. For example, C (*.c) displays all files with the .C extension. The drop-down box lists commonly used filename extensions.

You can also open a file by clicking the corresponding item in the project window. When you open a source file, its name is added to the Window menu. You can open only once a source file.

You can also open a file by dragging the file icon from Windows File Manager and dropping it into RIDE.

With RIDE, it is possible to open an included file directly from an edit window by selecting its path and choosing the **Open the Selected File** command from the pop-up menu.

7.2.5 Saving a file

To save a file:

1. Make the window corresponding to the file active.
2. Choose **Save** (Ctrl+S) from the **File** menu.
3. If it is a new file (untitled window) , RIDE displays the **Save As** dialog box. In the **File Name** box, type the filename.
4. In the **Drives** and **Directories** boxes, select a drive and directory.
5. Choose **OK**.

If the file already exists, the **Save** command saves changes without displaying the **Save As** dialog box. If the file is not changed RIDE displays a message box to notice that. Before saving the modifications of an opened source file, RIDE saves a copy of its content in a new file with the « .BAK » extension.

7.2.6 Printing files

If you have a printer driver installed or you have access to a network printer within Windows, you can print the entire content of the active Window.

You can select your printer and set its options in the **Printer Setup** dialog box, which you open by choosing **Printer Setup** on the **File** menu.

With RIDE , it is not possible to print only a part of source files.

To print the entire contents of a source file:

1. Choose **Printer Setup** from the **File** menu. RIDE displays the **Printer Setup** dialog box. Choose the printer you want to print with, and the other options of the printing.
2. Choose **Print** from the **File** menu. RIDE displays the **Print Options** dialog box.
3. Choose **Include Line numbers**, if you want to insert the line numbers in front of each line.
4. Choose **Include Page number**, if you want to insert the page number on the top of each page.
5. Specify the distance between the left edge of the page and the beginning of each line from the **Left Margin** control.
6. Choose **OK**.

7.2.7 Exiting from RIDE

This command exits the RIDE and removes it from memory.

If you have modified a source file without saving it, the RIDE prompts you to do so before exiting.

7.3 Editing a Text

7.3.1 Fast moving in the source file »

To move to a specific line with the file:

1. Click the right button of the mouse to display the pop-up menu of the view.
2. Select the command **Go To Line**. The **Go To** dialog box appears.
3. Type your selected line number.
4. Choose OK.

7.3.2 Using OEM characters »

RIDE uses only the ANSI character set. But it provides the possibility to convert characters from OEM to ANSI when loading the source file from disk and convert them back to OEM when saving the source file to disk. In this way, the user could view its file indiscriminately from MS-DOS and RIDE.

To select this option:

1. Select **Project** from the **Options** menu. The **Options** dialog box appears.
2. Select **Environment|Editor** from the Options list box.
3. Select **Convert OEM to ANSI** check box .
4. Choose **OK**.

7.3.3 Using Tabulation characters

RIDE supports tab stops in a source file. To set the number of spaces a tab is equivalent of:

1. Select **Project** from the **Options** menu. The **Options** dialog box appears.
2. Select **Environment|Editor** from the Options list box.
3. Modify the value of **Tab Stops**.
4. Choose **OK**.

7.3.4 To transfer text

To transfer a selected text from one editing window to another or from one place of the same editing window to another, you can use the Clipboard. To do this you can:

1. Select the text you want to transfer.
2. From the **Edit** menu choose **Cut** (Ctrl+X).
3. Place the insertion point in any editing window where you want to insert the text.
4. From the **Edit** menu choose **Paste** (Ctrl+V).

Cut and **Copy** » commands copy the selected text in the Clipboard. If you want to cancel the transfer, you can choose the **Undo** command (Ctrl+Z) from the **Edit** menu.

To delete an entire line , you can use the Ctrl+Y shortcut, but this command does not copy the text into the Clipboard.

7.3.5 To copy text

To copy a selected text from one editing window to another or from one place of the same editing window to another, you can use the Clipboard. To do this you can:

1. Select the text you want to transfer.
2. From the **Edit** menu choose **Copy** (Ctrl+C).
3. Place the insertion point in any editing window where you want to insert the text.
4. From the **Edit** menu choose **Paste** (Ctrl+V).

The **Copy** command copies the selected text in the Clipboard. If you want to cancel the transfer, you can choose the **Undo** command (Ctrl+Z) from the **Edit** menu.

You can copy text from another editor to RIDE and from RIDE to another editor, with using the Clipboard.

7.3.6 To undo an editing action

You can use the **Undo** command (Ctrl+Z) on the **Edit** menu to "undo" previous editing actions. The number and scope of editing actions you can undo is determined by the Size of the Undo buffer.

Undo inserts any characters you deleted, deletes any characters you inserted, replaces any characters you overwrote, and moves your cursor back to a prior position. If you undo a block operation, your file will appear as it was before you executed the block operation.

The Undo command will not change an option setting that affects more than one window or reverse any toggle setting that has a global effect; for example, Ins/Ovr.

This command is available only if an Edit window is currently active and there is something to undo.

RIDE, lets the user select the maximum number of the undo command available for en edit Window. To do so:

1. Select **Project** from the **Options** menu. The **Options** dialog box appears.
2. Select **Environment|Editor** from the Options list box.
3. Type the **Undo Number** of your choice.
4. Choose **OK**.

7.4 Finding and replacing text

7.4.1 To find a text string

To find a text string in a source file:

1. Set the cursor where you want to start your search.
2. If you select a word by double clicking the mouse left button the editor uses this word as a default search string.
3. From the **Search** menu, choose **Find**. The **Find** dialog box appears.
4. Select the string that you want to search. The editor retains the selected string between uses of the **Find** command in any single session.
5. If necessary, select one or more find options. (Find a whole word match: This "find" matches all occurrences of a text string not preceded or followed by an alphanumerical character or the underscore (_). Find a case-sensitive match: This "find" searches for text that matches the capitalization of the text string.)
6. Click **Find** to begin the search.

When you perform this type of search, each occurrence will be selected in the editing window, to find the next occurrence of the string you can click **Find** again or close the **Find** dialog box and use the **Next** Command (F3) from the Edit menu.

You can continue your search in the same window or select another one and search for the same string.

You can use the **Matching Bracket** (ALT+()) to search for a matching delimiter in an editing window.

7.4.2 To replace a text string

The Replace command searches the active window for a text string and replaces it with another text string.

Replace options are the following:

- Find a whole word match. This "find" matches all occurrences of a text string not preceded or followed by an alphanumerical character or the underscore (_).
- Find a case-sensitive match. This "find" searches for text that matches the capitalization of the text string.

To replace a text string:

1. Position the insertion point where you want to start your search.
2. From the **Search** menu, choose **Replace**. The **Replace** dialog box appears.
3. Type the search text or a regular expression in the **Find What** box.
4. Type the replacement text in the **Replace With** box.
5. Select any of the **Replace** options.

Begin replacing text by choosing **Find Next** or **Replace All**. If you choose **Find Next** the editor will prompt you to ask if you want replace this occurrence of the string. With choosing **Replace all** the editor will replace all the occurrence of the word without asking your permission.

7.5 Window management

Most of the windows in RIDE have all the standard window elements, such as scroll bars, Minimize and Maximize buttons, and a Control menu box. To open windows, use the **File|Open** or **File|New** commands.

The window management commands are:

Cascade

This command stacks all open windows and overlaps them so that each is the same Size as all others and only part of each underlying window is visible.

Tile Horizontal

This command arranges the RIDE windows from top to bottom so that they cover the entire width of the desktop without overlapping one another. If there are more than three open windows, they are arranged in a manner that allows more width than height.

Tile Vertical

This command is identical to Tile Horizontal but windows are arranged from left to right.

Tile Debug (available in debugging mode)

The Tile Debug command organizes the open windows of your actual debugging session.

Refresh (available in debugging mode)

The Refresh command refreshes the open windows data value (as well as watched expressions value) during the execution of the program.

Arrange Icons

This command rearranges the desktop icons.

Close All

Close all the opened windows.

List of open windows

RIDE displays a list of currently open windows at the bottom of the Window menu. When you choose an open window, you make that window active. A check mark appears in front of the active window.

7.6 « Help »

You can press F1 from within any RIDE Help window to activate the Microsoft Windows Help file. There, you will find information on using Microsoft Windows Help as well as information about moving, scrolling, or sizing windows.

In addition to Microsoft Windows Help commands, RIDE Help gives you access to:

Context-Sensitive Help (F1).

You can get information about the RIDE interface with using the context-sensitive help. To use the context-sensitive help:

- Place the focus on the desired element (menu, menu command, or item on a dialog box).
- Press F1.

Keyword Search (F1 in the Edit Window)

You can have direct access to Help about a term in your program. To activate the Keyword Search select the term you want to get help about, in your editing window. Then display the help window with pressing F1. You can also do that with choosing Search in the Help menu.

If Help is not available for the term, the RIDE Search windows appears. Press the Show Topics buttons or double-click on the term to display the list of topics containing that terms in the bottom list box. Press Go To or double-click on the topic title to view that topic.

Help Buttons (on dialog boxes)

To get help about different dialog boxes you can use the dialog's **Help** button to display the help topic corresponding to this dialog box.

Help on error message

To display more information about an error occurred during compilation, you can select the error item in the Message window and then press F1 to display the corresponding topic.

7.7 commands summary »

Commands	Shortcut	Button	Description
New	Ctrl+N		Create a new source file
Open	Ctrl+O		Open an existing source file
Save	Ctrl+S		Save the file of the active window.
Exit	Alt+F4		Exit from RIDE
Undo	Ctrl+Z		Undo the last editor command.
Cut	Ctrl+X		Remove the selected text and copies it to the clipboard
Copy	Ctrl+C		Copy the selected text to the Clipboard.
Paste	Ctrl+V		Paste from the clipboard in the position of the cursor.
Cut the current line	Ctrl+Y		Cut the editor line without copying the text into the clipboard.
Find			Find the specified text
Next	F3		Repeat the last Find or Replace command.
Next Message	Alt+F8		Find the next error position.
Previous Message	Alt+F7		Return to the last error.
Matching Bracket	Alt+(		Set the cursor position to the corresponding bracket.
Add File	Alt+Ins		Add a new file to the project.
Delete File	Alt+Del		Remove the selected file from the project.
Compile	Alt+F9		Compile or assemble the edited file.
Make	F9		Make the target of the current project.
Build all	Shift+F9		Rebuilds all the files in the project, regardless of whether they are out of date.
Tile			Tile open windows
Debug			Activates the integrated debugger (or another tool)

8. INDEX

Arrange icons,63

cascade,63

Compilation

 innovation,51

Copy,60

Debugging

 options,50

Debugging configuration,50

Editing a text,59

Editor

 options,56

Environment options

 editor,45

 execution,46

Environment options (32-bit)

 colors,44

 directories,43

 font,44

exiting from RIDE,58

F1,64

file,26, 57

 add to a project,28

 remove from a project,28

General presentation,9

go to line,59

Help,64

IDE,8

Installing RKit,12

Invocation,15

library manager,29

Listing file

 view,30

MAP file

 view,29

MA-ST6 options

 listing,38

 object,39

 set,37

 source,37

menu commands,20

Messages window

 view,30

new file,57

OEM,59

opening a file,57

Other tool with RIDE,50

Paste,59, 60

Presentation of the documentation,10

printing a file,58

project

 Build all,53

 compilation,52

 Link,53

 make,53

 modifying,28

- what is a project ?,24
- project
 - Creating a project,26
 - opening,28
- Project
 - options,31
- Project menu,26
- Project window,24
- RC-ST6 options
 - code generation,33
 - defines,34
 - listing,34
 - memory model,35
 - messages,36
 - object,35
 - optimizer,36
 - source,33
- refresh,63
- RKit-ST6,9
- RLST6 options
 - linker,40
- RL-ST6 options
 - bank switching,42
 - listing,41
 - more controls,41
- Running a tool,29
- saving a file,58
- Shortcuts,65
- status bar,21
- string
 - find,61
 - next,61
 - replace,62
 - replace all,62
 - search,61
- System requirements,12
- Target options,32
- tile debug,63
- tile horizontal,63
- tile vertical,63
- Tool bar,20
- Tools
 - options,47
- Tools
 - menu,29
- Undo,60
- window management,63