

Solaris Native Languages Support Quick Start Guide 1

This guide shows how to use the Solaris Native Languages Support features to develop C, C++, and Fortran programs.

To start the IDE, type `runide.sh` in a terminal window.

1. Accessing a Directory

Before you can access a directory from the IDE, you must first mount the directory as a filesystem (make the filesystem available to the IDE).

1. Click **Editing** in the main window to switch to the editing workspace and to display the Explorer window.
2. In the Explorer, click the **Filesystems** tab.
3. Right-click the **Filesystems** icon that is at the top of the Explorer window to display the contextual menu.
4. If you want to use one of the supported version control tools, choose **Mount** → **Version Control** from the contextual menu. Otherwise, choose **Mount** → **Local Directory**.

2. Creating a Source File

1. In the Explorer window, click the **Filesystems** tab.
2. Navigate to the directory where you want to put the file.
3. Right-click the directory to display the contextual menu. Then choose **New**.
4. Find and choose the file type.
5. Type the file's name without the extension (suffix) and click **Finish**.

The file appears in the editor.

Tip - If you do not find the file type that you need, create a text file and provide the desired extension.

3. Accessing a Source File

1. In the Filesystems tab in the Explorer, navigate to the source file.
 2. Double-click the file to open it in the editor.
- Tip** - The Core IDE online help explains how to use word match, macros, abbreviations, syntax highlighting, formatting shortcuts, and other editing features.

4. Creating a Makefile

The makefile tells the IDE how to compile your program. To create a makefile, choose **Tools** → **Makefile Wizard** from the main window.

5. Building an Executable

1. In the Explorer, navigate to the makefile.
2. (Optional) To remove object files from the prior build, right-click the makefile and choose **Clean** from the contextual menu.
3. (Optional) Use the makefile's Properties window to specify a target or to pass arguments to the `make` command.
4. Right-click the makefile and choose **Execute** from the contextual menu.

The IDE displays make messages and compiler messages in the Output window.

Tip - See the online help for information about using `make`.

6. Running a Program

1. In the Explorer, navigate to the executable.
2. (Optional) In the Execution tab in the Properties window, type the arguments that you want to pass to the executable.
3. Right-click the executable and choose **Execute** from the contextual menu.

The program's output appears in the Output window.

Solaris Native Languages Support Quick Start Guide 2

7. Debugging a Program

1. If you are debugging a Java class, open the class file's Properties window and set the Default Debugger property to **dbx Debugger**.
2. In the Explorer, right-click the executable and choose **Debug**→**Start** from the contextual menu.
3. If a debugging session is running, the Start New Session Dialog box appears. To terminate the current session and start a new session, click **Finish & Start**. To add a new session, click **Start Additional Session**.

A Debugger window appears.

For information about using the Debug window, see the Debugging With the dbx Debugger topic in the Solaris Native Languages Support online help.

Tip – To attach a debugging session to an active Solaris™ process, choose **Debug**→**Attach to Solaris Process** in the main window. To debug a core file, right-click the core file in the Explorer window and choose **Debug**→**Load Program** from the contextual menu.

8. Analyzing Program Performance

1. **Compile** your C program, C++ program, or Fortran program using the `-g` compiler option.

You do not need to compile Java™ programs.

2. **Collect** performance data for your program.

Use the `collect(1)` command to create an **experiment** file. The syntax of this command is as follows:

```
% collect collect-options prog-name prog-args
```

For Java programs, you must use the `collect` command with a version of the Java 2 SDK that is no earlier than 1.4. The `collect` command recognizes files with the `.jar` and `.class` extensions.

Tip – For best results, use the `collect` command on a computer that is not running other programs.

3. In the Explorer window, navigate to the **experiment**. Double-click the experiment to start the Performance Analyzer tool.

For more information about analyzing program performance and about using the tool, see the online help in the Performance Analyzer tool.

9. Getting Help

To Get Help About...

Do This

Starting the IDE

See the **Release Notes**

Using the IDE to **create, compile, debug,** and **analyze** your **C** programs, **C++** programs, and **Fortran** programs

Choose **Help** → **Helpsets** → **Solaris Native Languages Support Help**.

Using the **basic IDE features**

Choose **Help** → **Help Sets** → **Core IDE Help**.

Using a **window** in the IDE

Click in the window. Then press **F1**.

Accessing the Sun™ Open Net Environment (Sun ONE) Studio Compiler Collection **man pages** (including API reference information), **manuals**, and **readmes**

Open the following URL in a web browser:

```
file:/opt/SUNWsp/docs/index.html
```

If your Compiler Collection **software is not installed** in the `/opt` directory, ask your system administrator for the equivalent path on your system.