Getting Started with Maple IDE

Getting Started with Maple IDE

Copyright

This document was produced using Maple and DocBook.

Contents

Introduction

Maple IDE provides you with features that help you develop Maple™ code quickly and efficiently.

Features of this toolbox include:

- Powerful Maple code editor
- Syntax highlighting
- Automatic indenting
- Source code validation
- Syntax checking with Mint (Maple installation required)
- Maple library (.mla) creation (Maple installation required)
- Code templates to quickly generate commonly used code
- The ability to rename modules, procedures, and variables
- The ability to build libraries using different Maple versions (Maple installation required)

Maple IDE is powered by technology from DigiArea Inc.

1 Configuring Maple IDE

After starting Maple IDE for the first time, you need to specify at least one workspace for your project files and (if you have Maple installed) at least one Maple Environment for your project to run in.

A workspace is a folder where Maple IDE places the resources for your Maple projects (that is, the source code files, Maple library archives, and project-specific settings).

A Maple Environment points Maple IDE to a Maple installation on your computer. Maple IDE uses the programs from your Maple installation to run your projects and check the syntax of your Maple source files. You can have as many Maple Environments as you have installed versions of Maple.

This chapter covers the following topics.

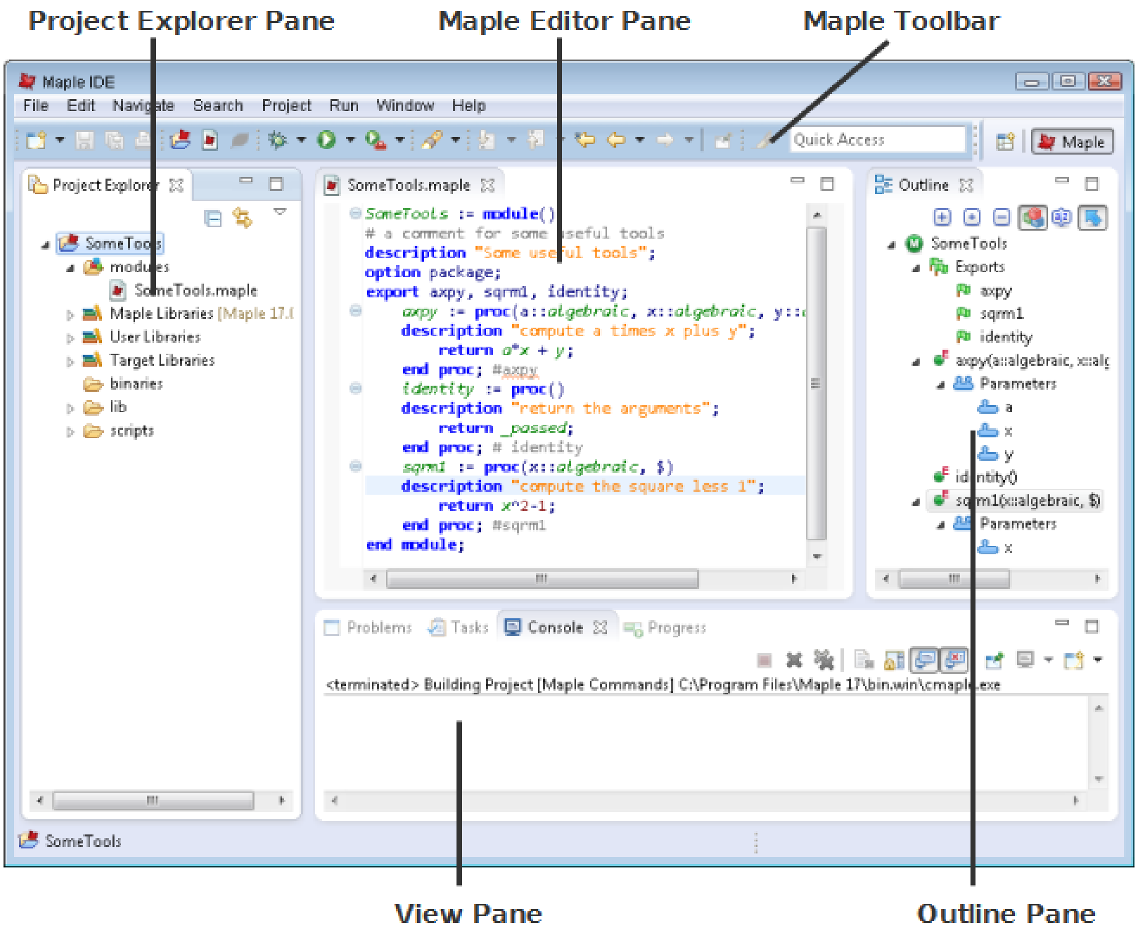
1.1 Starting Maple IDE for the First Time

To start Maple IDE:

1. Browse to the location where you installed Maple IDE.
2. Double-click on the Maple IDE executable.

   Note: On Windows, the executable is called MapleIDE.exe. On Macintosh and Linux it is called MapleIDE.
3. If you do not have any workspaces configured, the **Workspace Launcher** dialog opens. To choose your first workspace folder, do the following:
   a. Click **Browse...**.
   b. Navigate to, and then select the folder for your workspace.
   c. Click **OK** to close the Select Workspace Directory dialog.
   d. Click **OK** to close the Workspace Launcher dialog.
4. If Maple IDE starts with the welcome pane, click **x** to close the pane. (To return to the welcome pane, select **Help > Welcome**.)

Your Maple IDE window should look like the following figure.



```
SomeTools := module()
# a comment for some useful tools
description "Some useful tools";
option package;
export axpy, sqrm1, identity;
    axpy := proc(a::algebraic, x::algebraic, y::
    description "compute a times x plus y";
        return a*x + y;
    end proc; #axpy
    identity := proc()
    description "return the arguments";
        return _passed;
    end proc; # identity
    sqrm1 := proc(x::algebraic, $)
    description "compute the square less 1";
        return x^2-1;
    end proc; #sqrm1
end module;
```

```
<terminated> Building Project [Maple Commands] C:\Program Files\Maple 17\bin.win\cmaple.exe
```

The window contains the following features:

- **Maple Toolbar:** The Maple toolbar contains shortcuts for many common commands (for example, Run, Check Syntax, and Save). If the toolbar is not visible, select **Window > Show Toolbar**.
- **Project Explorer Pane:** The Project Explorer shows all the projects in your current workspace along with their associated folders and files.
- **Maple Editor Pane:** This Maple Editor pane contains all of the source code file you have open for editing.
- **Outline Pane:** The Outline pane gives you an overview of your code by showing you the structure of the modules, procedures, and parameters declared in the current source file.
- **View Pane:** The View Pane contains tabs that provide more information about the project. For example, the Console tab is where you can interact with your code using command-line Maple and the Tasks tab is where you can see all of the tab which lists all of **TODO** or **FIXME** tasks for your code.

1.2 Managing Your Workspaces

Workspaces provide a convenient way for you to organize similar projects. For example, if you are developing projects for different departments, you can group your projects by department by creating different workspaces for each department. You can add as many workspaces to Maple IDE as you want.

To add a workspace:

1. Select **File > Switch Workspace > Other...**.
2. Click **Browse**.
3. Navigate to, and then select the folder for your workspace.
4. Click **OK** to close the S**elect Workspace Directory** dialog.
5. Click **OK** to close the **Workspace Launcher** dialog.

After creating the workspace, Maple IDE restarts in the new workspace.

To switch to a different workspace

1. Select **File > Switch Workspace**.
2. Select the workspace from the list of available workspaces.

Maple IDE restarts in the workspace you selected.

1.3 Configuring Maple Environments

If you have Maple installed, Maple IDE can check the syntax, run, and generate a Maple library archive (.mla) file for your project. In order to do these tasks, you must add a Maple Environment in Maple IDE that corresponds to your Maple installation. You can have as many Maple Environments as you have installations of Maple.

To add a Maple Environment:

1. From the menu bar, select **Window > Preferences**.
2. Select **Maple > Environments**.
3. Click **Add**.
4. Click **Browse**.
5. Navigate to the location of your Maple installation folder, select it, and then click **OK**.

   Note: If you do not know where Maple is installed, start Maple, and then run the kernelopts(mapledir) command. The output of kernelopts(mapledir) is the location of the Maple installation folder.

   Maple IDE automatically initializes the environment by populating the remaining fields in the **Add Maple Environment** dialog. These fields cannot be edited.
6. Click **Add**.

The new Maple Environment is listed in the **Environments** section. If you have more than one environment listed, the selected environment is used by default by Maple IDE.

1.4 Next Steps

Chapter 2 of this guide creates a Maple project called SomeTools. It covers creating the new project, adding a file with your Maple source code, checking the syntax for your source code, running the project, and generating the .mla file for your project.

2 Creating and Running a Project - SomeTools

This chapter goes through the creation of a new project called *SomeTools*. The project is a module that has the package option and contains three exported procedures (axpy, identity, and sqrm1). The chapter concludes with a section on how you can get additional help on using Maple IDE.

The following topics are covered in this chapter.

2.1 Creating the SomeTools Project

To create the SomeTools project:

1. Select **File > New.. > Maple Project**.
2. In the **Project name** text box, enter **SomeTools**.

   Note: Do not include spaces or special characters (for example, \ or &) in the project name.
3. Click **Finish**.

The *SomeTools* project appears on the **Project Explorer** pane. The project contains folders for storing your source code (*modules*) and your Maple library archive files (*binaries*).

2.2 Setting Project-Specific Preferences to the SomeTools Project

If you have more than one Maple Environment defined in Maple IDE, you can specify which Maple Environment is used by the SomeTools project.

To assign a Maple Environment to your project:

1. Under the **Project Explorer** pane, right-click (**Control**-click for Macintosh®) the **SomeTools** folder, and then select **Properties**.
2. From the left-hand navigation, select **Maple > Build Path**.
3. Select the **Libraries** tab.

   This shows the Maple Environment currently being used to build this project.
4. Click **Add Environment**.
5. Select the Maple Environment you want to use for the project, and then click **OK**.

2.3 Adding Maple Files to Your Project

To add a Maple source file to SomeTools:

1. Under the **Project Explorer** pane, right-click (**Control**-click for Macintosh) the **SomeTools** folder, and then select **New > Maple File**.
2. Select **SomeTools > modules** as the parent folder for the Maple file.

3. In the **File name** text box, enter **SomeTools**, and then click **Finish**.

   Note: Do not include spaces or special characters (for example, \ or &) in the file name.

   A blank source code file named **SomeTools.maple** opens in the Maple Editor pane.

4. Enter the following code in **SomeTools.maple** :

```
SomeTools := module()

description "Some useful tools";

option package;

export axpy, sqrm1, identity;

axpy := proc(a::algebraic, x::algebraic, y::algebraic, $)

description "Compute a times x plus y.";

return a*x + y;

end proc; #axpy

identity := proc()

description "Return the arguments.";

return _passed;

end proc; # identity

sqrm1 := proc(x::algebraic, $)

description "Compute the square less 1.";

return x^2-1;

end proc; #sqrm1

end module;
```

5. From the **Maple Toolbar**, click **Save** (💾).

2.4 Syntax Checking with Mint

If you have Maple installed and have configured a Maple Environment for your project, you can use Mint to check the syntax in your Maple files.

To check the syntax in your Maple file:

1. From the **Project Explorer** pane, select **SomeTools > Modules > SomeTools.maple**.
2. From the **Maple Toolbar**, click **Check Syntax** (🍃).

The Mint syntax checker runs on your source code. The **Console** tab in the **View** pane shows the results from Mint.

2.5 Running Your Project

When you run your project, Maple executes the commands in your Maple file and displays the results in the **Console** tab on the **View** pane. You can test your code by entering commands in the **Console** tab.

To run your project:

1. From the **Project Explorer** pane, select **SomeTools**.
2. From the **Maple Toolbar**, click **Run** (▶).

The **Console** tab shows the results from running your code. You can enter commands to test your code. For example, if you enter the following commands in the **Console**

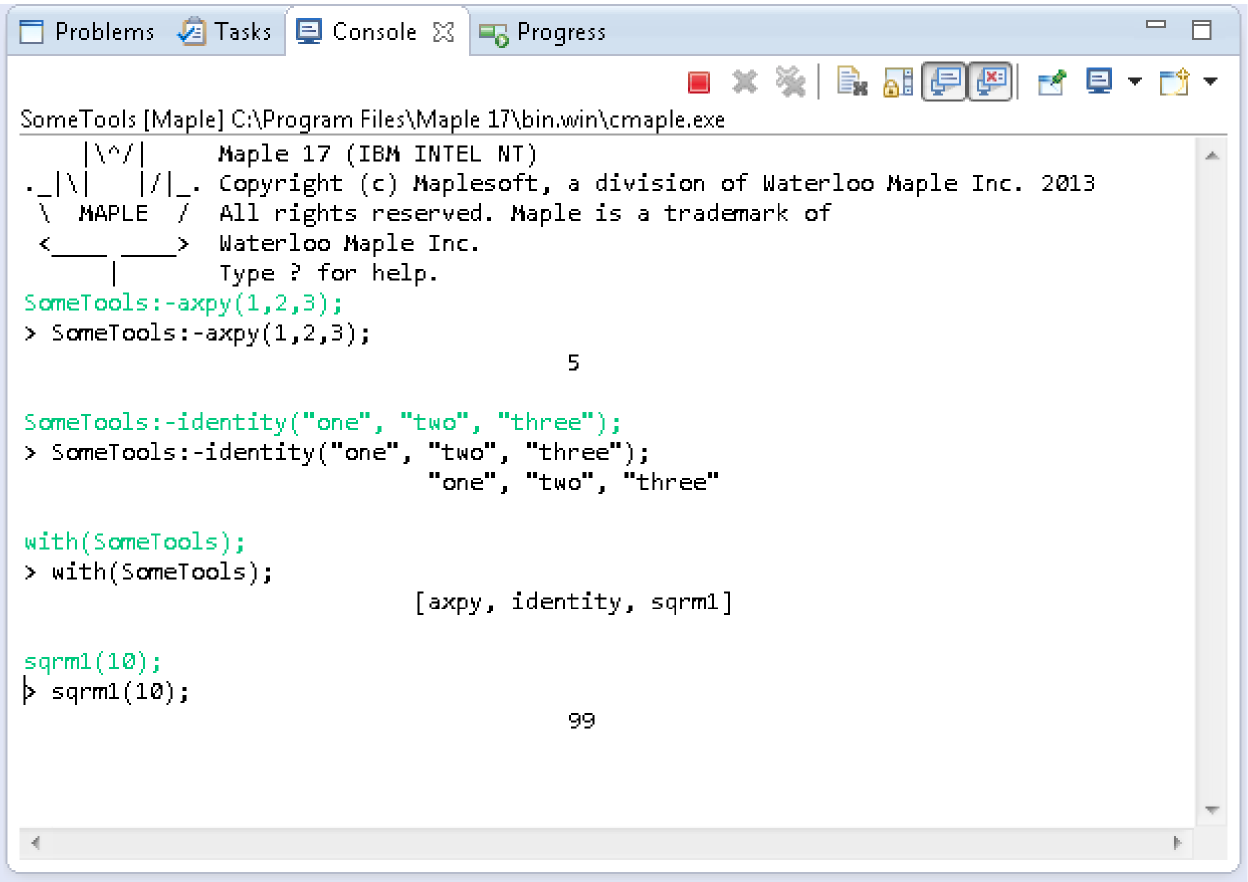
```
SomeTools:-axpy(1,2,3);

SomeTools:-identity("one", "two", "three");

with(SomeTools);

sqrm1(10);
```

you should get the same output as in the following figure.



SomeTools [Maple] C:\Program Files\Maple 17\bin.win\cmaple.exe

```
    |\^/|     Maple 17 (IBM INTEL NT)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2013
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
SomeTools:-axpy(1,2,3);
> SomeTools:-axpy(1,2,3);
                                        5

SomeTools:-identity("one", "two", "three");
> SomeTools:-identity("one", "two", "three");
                             "one", "two", "three"

with(SomeTools);
> with(SomeTools);
                            [axpy, identity, sqrm1]

sqrm1(10);
> sqrm1(10);
                                        99
```

2.6 Using Your Library in Maple

After running your project, Maple IDE generates a Maple library archive (.mla) file and saves it in the **binaries** folder of your project. As shown in the previous section, Maple IDE is configured to look for .mla files in the binaries folder, and this is why you can use the package and procedures you defined in your project in the **Console**.

Your Maple installation, however, is not configured to look for library files in your project's **binaries** folder. In order to use your library with your installed version of Maple, you have to copy the generated .mla file to a folder in which Maple looks for libraries. Maple automatically looks for libraries in the **lib** subdirectories of directories in

HOMEDIR/maple/toolbox. Copying your .mla file to an appropriate **lib** subdirectory makes your library available to you in Maple.

To use your library in Maple:

1. Go to the **HOMEDIR/maple/toolbox** directory on your computer.

   Note: HOMEDIR is your home directory. If you are not sure what your home directory is, start Maple, run the kernelopts(homedir) command, and then use the output of this command for **HOMEDIR**.

2. Create a directory called **SomeTools**.
3. Go into the **SomeTools** directory, and then create a directory called **lib**.
4. From the **Project Explorer** pane in Maple IDE, expand the **SomeTools > binaries** folder.

   Note: If you cannot expand the **binaries** folder, right-click (**Control**-click for Macintosh) on **binaries**, and then select **Refresh**.

5. Copy the **SomeTools.mla** file into **HOMEDIR/maple/toolbox/SomeTools/lib**.
6. Start Maple.
7. At the Maple prompt, enter the following commands:

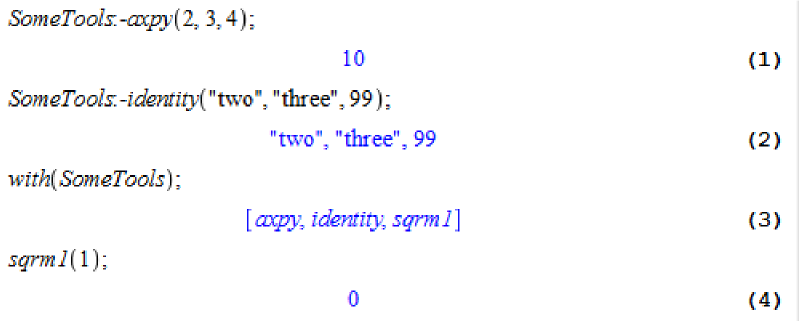
   ```
   SomeTools:-axpy(2,3,4);

   SomeTools:-identity("two", "three", 99);

   with(SomeTools);

   sqrm1(1);
   ```

   You should get the same output as in the following figure.



```
SomeTools:-axpy(2, 3, 4);
                                 10                               (1)
SomeTools:-identity("two", "three", 99);
                          "two", "three", 99                      (2)
with(SomeTools);
                        [axpy, identity, sqrm1]                   (3)
sqrm1(1);
                                  0                               (4)
```

2.7 Getting Maple IDE Help

Maple IDE comes with its own help system. You can find information on more advanced topics by accessing the Maple IDE help system.

To access the Maple IDE help system:

1. Select **Help > Help Contents**.
2. Expand the **Maple Programming User Guide** entry to browse through the Maple IDE topics. Alternatively, you can search for a topic using **Search**.

In addition to the help system, you can find more resources (for example, videos, tutorials, and forums) related to Maple IDE on these web sites:

- The DigiArea Maple IDE product page (http://www.digi-area.com/CasStudio/Maple/)

- Maple Primes (http://www.mapleprimes.com)