Hypercosm™
We Put 3D To Work!℠

Hypercosm Studio

www.hypercosm.com

Hypercosm Studio Guide

Revision: November 2005

Hypercosm LLC
(608) 827-5000
www.hypercosm.com

For technical support, please contact Hypercosm by their web site, the phone number above, or by email at support@hypercosm.com

To see support options, frequently asked questions, and other information, please see the Hypercosm support web site at www.hypercosm.com/support/

Table of Contents

Introduction

Welcome to Hypercosm Studio! Studio is Hypercosm's 3D graphics programming environment for web developers. It contains all the tools you need to created Hypercosm 3D applets and to integrate those applets into your HTML files.

With Hypercosm Studio you can:

- Create and edit programs in the OMAR programming language
- Compile those programs into Hypercosm applets
- Create a sample web page to test your applets
- Modify existing HTML files to include your Hypercosm applets (either automatically or by hand)
- Manage your HTML, OMAR, sound, and texture files together as complete projects

This guide describes how to use Hypercosm Studio to create programs in the OMAR programming language, compile those programs, and include them in your web pages. It does not describe the specifics of the OMAR programming language; for that information you need to see *The OMAR Programming Language Reference Manual and Programming Guide*, and *The Hypercosm 3D Graphics Systems Guide*.

Getting Started

This section takes you through the steps you should follow before you start using Studio. It also introduces useful information about the different files you'll be using.

Installing Studio

When you are ready to install Studio, you should exit all other Windows programs, and then run **Setup.exe** on the Hypercosm Studio CD. The install program guides you through the rest of the installation process. It is highly recommended that you install the Hypercosm 3D Player (which allows you to view Hypercosm applets through a web browser), and Adobe Acrobat Reader (if you do not already have Acrobat Reader). You need Acrobat Reader to view the electronic versions of Hypercosm's manuals.

The setup program adds Hypercosm Studio to your Windows **Start > Programs > Hypercosm** menu. From this menu you can also open up the electronic versions of the *Hypercosm Studio User Guide* (this manual), the *OMAR Programming Language Reference Manual and Programming Guide,* and the *Hypercosm 3D Graphics System Guide*.

Overview of Hypercosm File Types

OMAR files, composed of OMAR programming code, contain all of the definitions, instructions, and data that are needed to create Hypercosm 3D applets. There are two different OMAR file types, both of which can be created and edited using Studio:

- **OMAR source files** are runnable files that contain essential instructions for producing Hypercosm applets. Source file names carry the extension **.omar**.
- **OMAR resource files** are not runnable, but contain pieces of OMAR code that may be imported for use in source files. Resource file names carry the extension **.ores**.

Both the OMAR file types are described in more detail in the *OMAR Programming Language Reference Manual and Programming Guide* and the *Hypercosm 3D Graphics System Guide*. In addition to the OMAR file types, Hypercosm Studio also creates the following file types:

- **Hypercosm Studio project files** keep track of all the different files in a single Studio *project*. Projects are discussed in a later section of this guide. Studio project files carry the extension **.hsp**, which stands for "**H**ypercosm **S**tudio **p**roject."
- **Hypercosm applets** are files that can be included by HTML files and run in web browsers. Hypercosm applets carry the extension **.hcvm**, which stands for "**H**ypercosm **v**irtual **m**achine," and are often called HCVM files. Hypercosm applets are described in more detail in the next section.

Hypercosm Applets

The production of Hypercosm applets is the ultimate goal of most Studio usage. Only by creating Hypercosm applets are you able to present Hypercosm 3D graphics on your web site. For that reason it is important to understand how applets work, and to know about the different kinds of applets that can be created.

How Applets Work

To produce 3D graphics with the Hypercosm systems, you need to write programs in OMAR, and then run them with a Hypercosm *interpreter*. *Interpreted* programming languages, such as OMAR and Java, are generally similar to conventional programming languages, such as C, Pascal, and FORTRAN, but differ in their implementation.

To run programs on a computer, programming languages must somehow be converted, or *compiled*, into *machine language*—the 1's and 0's that computers actually understand. C, Pascal, and FORTRAN programs are all compiled directly into machine language *executable code* that runs by itself. Because the resulting executable file is written in one particular machine language, it cannot be run on all kinds of computers, but only on computers that understand that particular machine language.

When you use OMAR or Java, however, your code is first compiled into a generic intermediate code, or *byte code,* that is not specific to any particular type of machine. To run this byte code on your computer, it must be run through a special program called an *interpreter,* or *virtual machine,* which translates the byte code into machine language instructions that can be executed on your particular computer, no matter what machine language your computer understands.

Hypercosm applets are files composed of OMAR byte code. You create applets by running your OMAR code through the Studio compiler. How you can do this is described in greater detail in later sections. Once you have created an applet, you can include it in a web page (by following instructions given in this guide) and view the applet in a web browser.

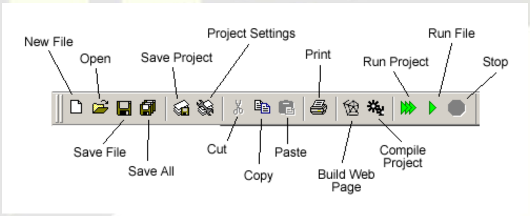
To view Hypercosm applets in web browsers, you must have the Hypercosm 3D Player, which acts as an interpreter, or virtual machine. The 3D Player runs the Hypercosm byte code by translating it into instructions that your particular machine can understand. In this way, you can put one Hypercosm applet on a web page and be assured that it will run correctly on any kind of machine, so long as that machine has a Hypercosm 3D Player.

Writing OMAR Source Code

The first step in creating a Hypercosm applet is to write OMAR source code that describes your applet. You can start working with OMAR files by examining the sample files that are included with Studio. Running them is a good way to see the range of programs you can produce with OMAR. Reading and editing the sample code is a good way to learn about various programming techniques.

To open one of the sample files, use the **File** menu and select **Open**. By default, the sample files are placed in **C:\Program Files\Hypercosm\Hypercosm Studio\Examples** when you install Studio. You can go to this directory by clicking on the **Examples** button at the bottom of the **Open** window.

Anyone who has used a Windows text editor or word processor will immediately be familiar with Studio's interface. Here's a quick look at Studio's toolbar:

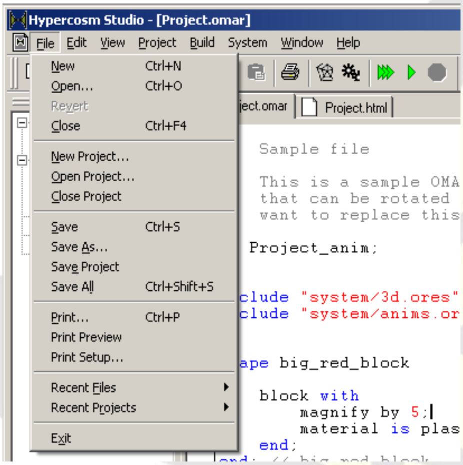


Studio's commands are standard Windows-based editor commands.

To open a new or existing file, use the **File** menu and select **New** to create a new file, or **Open** to open an existing file.

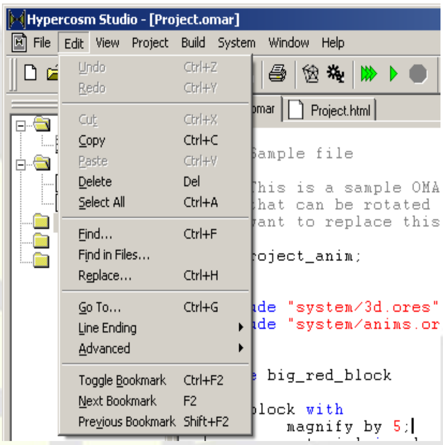
Standard File Management and Editing Features

Hypercosm Studio provides the standard Windows management and text editing features. The file management features are located under the **File** menu, and include the commands you'd expect to use when creating, opening, saving, and printing files.



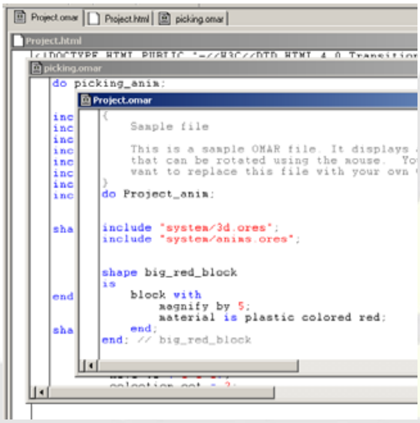
```
  Sample file

  This is a sample OMA
  that can be rotated
  want to replace this

 Project_anim;

clude "system/3d.ores"
clude "system/anims.or

ape big_red_block

  block with
      magnify by 5;
      material is plas
  end;
```

Once you’ve created or opened a file, you can use the text editing features under the **Edit** menu. They include commands for undoing, using the clip-board, and selecting and finding text.



```
ample file

his is a sample OMA
hat can be rotated
ant to replace this

oject_anim;


de "system/3d.ores"
de "system/anims.or


 big_red_block

lock with
    magnify by 5;
```

When you are creating or editing an OMAR program, the work takes place in a window. You can have multiple windows open at one time.



```
{
    Sample file

    This is a sample OMAR file. It displays
    that can be rotated using the mouse.  Yo
    want to replace this file with your own
}
do Project_anim;

include "system/3d.ores";
include "system/anims.ores";

shape big_red_block
is
    block with
        magnify by 5;
        material is plastic colored red;
    end;
end; // big_red_block
```

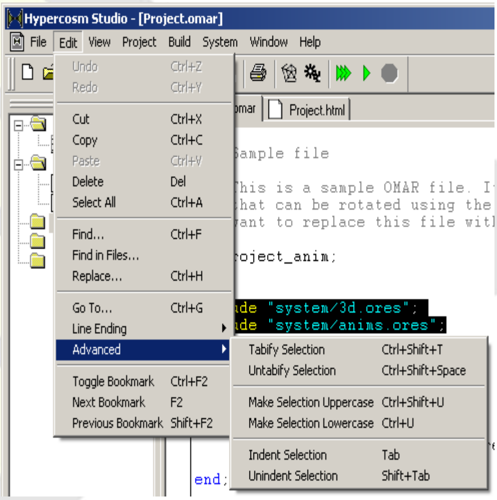
You can specify whether or not you can see the left margin using the window under **View** > **Options** > **Editor.** You can also tell Studio to remember which files you had open, and where on the screen they were placed when you exited from Studio. The next time you start up Studio, it opens those files and places them exactly where they were, for your editing convenience. Use the checkbox **Remember open documents and their positions** to control this feature.

The **View** menu lets you see or hide the **Toolbar**, **Status bar**, **Output**, and **Project Manager**. In most cases you should always display the toolbar and the status bar unless you really need the space on your screen.

Code Editing Features

In addition to standard editing features, Hypercosm Studio provides a few extra features designed for creating and editing code. Lines of code have somewhat different formatting conventions than standard text; these conventions include regular indentation, tab settings, and white space handling.

Studio isn't picky about how your file looks. You can use upper or lowercase letters. You can separate words with tabs or spaces. You can indent wherever you like. Then you can change these things with the features on the **Edit > Advanced** menu.



Hypercosm Studio - [Project.omar]
File Edit View Project Build System Window Help
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Select All Ctrl+A
Find... Ctrl+F
Find in Files...
Replace... Ctrl+H
Go To... Ctrl+G
Line Ending
Advanced
Toggle Bookmark Ctrl+F2
Next Bookmark F2
Previous Bookmark Shift+F2
Tabify Selection Ctrl+Shift+T
Untabify Selection Ctrl+Shift+Space
Make Selection Uppercase Ctrl+Shift+U
Make Selection Lowercase Ctrl+U
Indent Selection Tab
Unindent Selection Shift+Tab
omar
Project.html

```
Sample file

This is a sample OMAR file. I
hat can be rotated using the
ant to replace this file wit

oject_anim;

ude "system/3d.ores";
ude "system/anims.ores";

end;
```

- Choose **Edit > Advanced > Make Selection Uppercase** to change a selected portion of text to all uppercase. Use **Make Selection Lowercase** to change a selected portion to lowercase.
- Choose **Edit > Advanced > Indent Selection** to move a selected portion of text over one tab stop. Use **Unindent** to move the selection one tab stop in the other direction.

To display all spaces and tab markers, choose **View > Whitespace.**

Tabs

You can specify the number of columns per tab stop. The default setting is 4 columns, that is one tab every four columns. Choose **View > Options > Formatting > Tabsize** to change the number of columns per tab stop.

Whenever you want, you can change tabs to spaces, and spaces back to tabs. Choose **Edit > Advanced > Tabify Selection** to change all white space to tabs. Note that Studio only changes spaces to tabs when there are as many spaces as you have set tab settings. For example, if you are using the default setting of four columns per tab stop, Studio changes groups of four or more spaces to tabs, but leaves as spaces groups of three or fewer spaces. **Edit > Advanced > Untabify Selection** changes all tabs to spaces.

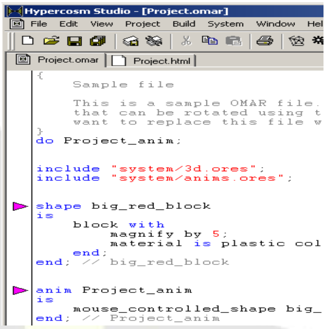
Why would you want to change spaces to tabs and back again? In most cases, you don't need to. However, if you want to view your files with a different text editor or word processor with different ideas about how big a tab should be, you might want to change hard spaces to tabs or vice versa.

Moving Around in Your File

You can move to a specific line in your file using the **Go To** command on the **Edit** menu. Choose **Go To** and enter a line number.

You can see the line numbers in a file by dragging the elevator box in the scrollbar. When you drag the elevator box, Studio pops up a little box that shows the line number of the line at the top of the window. You can turn this feature off by disabling the check box **Show line number tooltip while scrolling**, under **View > Options > Editor**.

You can set bookmarks in your files, and then move quickly to the spot marked by the bookmark. Bookmarks display as triangles in the margin.


Hypercosm Studio - [Project.omar]
File Edit View Project Build System Window Hel
Project.omar
Project.html

```
{
    Sample file

    This is a sample OMAR file.
    that can be rotated using t
    want to replace this file w
}
do Project_anim;

include "system/3d.ores";
include "system/anims.ores";

shape big_red_block
is
    block with
        magnify by 5;
        material is plastic col
    end;
end; // big_red_block

anim Project_anim
is
    mouse_controlled_shape big_
end; // Project_anim
```

- To set a bookmark, move to the line you want bookmarked, and choose **Edit > Toggle Bookmark**.
- To delete a bookmark, move to the line where the bookmark is located, and choose **Edit > Toggle Bookmark**.
- Each time you select **Next Bookmark** the cursor moves to the next bookmark in the file. Selecting **Previous Bookmark** moves the cursor to the previous bookmark in the file.

Colors and Fonts

Notice that when you create a file, OMAR reserved words and comments display in different colors from the rest of your text. This helps you distinguish different parts of your files more easily. You can customize the colors that Studio uses in the **View > Options > Editor Colors and Fonts** window. You can disable this feature entirely under **View > Options > Editor**.

Printing Your Files

Hypercosm Studio provides standard printing functions. You can use **File > Print Preview** to see a preview of your file, and **File > Print** to print it.

The Project

At the heart of Hypercosm Studio is the **Project**. A project consists of all the files needed to produce a single Hypercosm applet (including any sound or texture files that the applet uses). In addition, the project includes the specific HTML files or files that include the Hypercosm applet. Note that if you are creating multiple applets, they should be treated as multiple projects.

Hypercosm Studio won't interfere with the way your HTML files are already being managed, and won't required any changes to the way you are currently managing them; however, it does need to know where the file (or files) are, and what they are called. Only then can it integrate the Hypercosm applet into your HTML code. If you are starting from scratch, however, you can use Hypercosm Studio to create and manage these files and directories.

Studio provides a **Project Manager** that allows you to categorize and access all the files in your project. The Project Manager is a floating window that displays alongside Studio's working window. You can hide the project manager using the **View** menu, but it's more convenient to leave it visible.

Creating a Project

A project can be created at any time during the applet development process, but you must put all your files in a project sometime before you actually build the web page that will contain your applet. You may find it easier to begin coding an OMAR program without using a project in order to avoid the organization overhead that project requires. Later you can create a project for your OMAR program if you've decided you'd like to build a web page containing it.

You may, on the other hand, wish to create a project immediately when you start your applet development. This style of development makes sense when you know you'll eventually be building a web page for your applet, and you'd like to keep everything organized from the very beginning.

Studio supports both styles of development.

Creating a Project

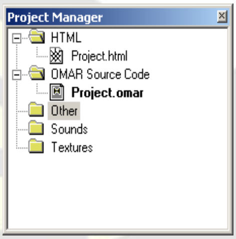
1. Click on **File**, then choose **New Project . . .**.

2. Decide on a name for the new project, and enter the name.

3. Choose a directory for the project.

4. Leave the checkboxes at the bottom of the **New Project** window at the default settings.

Studio creates the directory you specified, and creates some template files in that directory (more about them in a minute). It then shows your project in the Project Manager. The tree structure initially looks like this:



This Project Manager tree structure does not actually match the real directory structure. The categories you see are there only to aid in organizing the files you have in your project.

In the **HTML** category, Studio has created a template HTML file for testing purposes. You can use this template file when you want to see how your Hypercosm applet looks on an actual web page, or you can remove it and replace it with an existing HTML file that you've already created. (You can read how to do this later.)

Hypercosm Studio has also created a template OMAR file and some empty subcategories. The OMAR template file is named whatever you named your project, with the extension **.omar**. You can begin to create an OMAR program with this file, or you can ignore this file and start from scratch with a new file.

Notice the three remaining categories called **Other**, **Sounds**, and **Textures**. If you are creating an OMAR program that uses resource files like sound files (for example, in .wav format), you need to tell Studio about these files. If you don’t, Studio won’t be able to find them when the time comes to build the applet. A following section describes how to tell Studio about the location of your files.

Adding Files to Projects

If you are not starting a web page from scratch, and want to add a Hypercosm applet to an existing web page, you need to tell Studio where to find the HTML files (and you should then remove the template HTML file from the project). Similarly, if during the applet development process you include more sound, texture, or other files in your OMAR program, you should add them to your project, too.

Follow these directions:

1. Click on the **Project** menu and choose **Add Files**. Studio pops up a little menu.

   Or, you can right-click (that is, click with the right mouse button) on the tree in the Project Manager. When you right-click, Studio pops up a little menu. Click on **Add Files to Project**.

2. When Studio displays a file selection box, navigate to the directory that holds the file you want to add and select it.

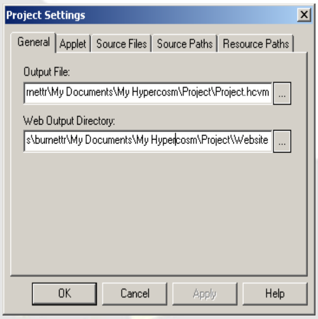
Studio now knows the location of the file you have just added, and treats it as part of the project. You can continue to add files in this manner. Note that files added *after* the project is created are *not* copied in to the project directory, but are used in their original location.

Working with Files in the Project Manager

The Project Manager has many features that make working with the files in your project easier. Right-clicking on any files in the Workspace pops up a little menu of options. Using the pop-up menu, you can open a file, or remove it from your project. If you right-click on an OMAR file, the pop-up menu also lets you run the file or set it as your main file (the file that is run when the project is run).

Project Settings

Selecting **Project > Settings** opens up the **Project Settings** window. This window allows you to set various attributes of your project, to work with files in your project, and to set various attributes of the applet the project will produce. The features of the **Project Settings** window are divided under five different tabs: **General**, **Applet, Source Files, Source Paths,** and **Resource Paths**.

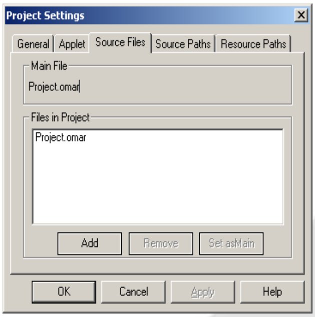


Output File:
rnettr\My Documents\My Hypercosm\Project\Project.hcvm

Web Output Directory:
s\burnettr\My Documents\My Hypercosm\Project\Website

Under the **General** tab, you can rename your project, or rename or relocate the output file (the applet file that is created when the project is compiled).

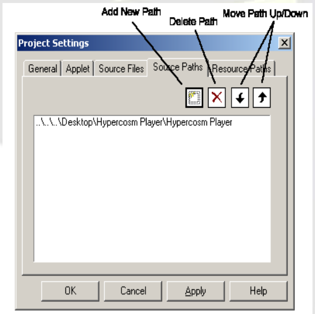
You can also relocate the web output directory (the directory where Studio places the HMTL it produces, along with any other files needed to produce an applet embedded in a web page). *Note: you can also use relative paths for your Output Files and Web Output Directory (e.g. ../../Applet/Website).*



Main File: Project.omar

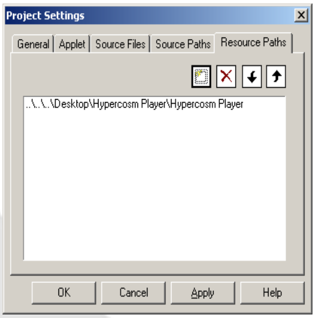
Files in Project: Project.omar

Under the **Source Files** tab, you can add files to and remove files from the project. You can also set the main OMAR files (the one that is run when Studio runs the project).



..\..\..\Desktop\Hypercosm Player\Hypercosm Player

Under the **Source Paths** tab, you can add relative paths to other **.omar** and **.ores** files that are used by your applet through an ‘include’ command. This is useful if you create your own library of **.omar/.ores** files that you regularly include in your applets



..\..\..\Desktop\Hypercosm Player\Hypercosm Player

Under the **Resource Paths** tab, you can add relative paths to resources that are used by your applet; e.g. textures and sound files. This is useful if you have your own library of textures and sounds that you regularly include in your projects

Note: Source and Resource Paths can be set for the entire system so it can be used for all of your applets and projects by going to ***System->Settings->Source Paths*** *or* ***System->Settings->Resource Paths****.*

The features under the **Applet** tab are discussed in the next section.

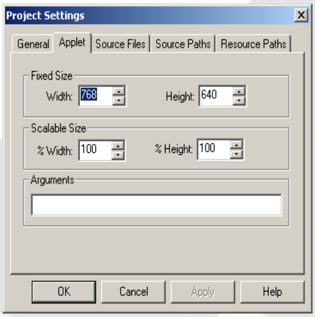
Applet Settings & Program Arguments

To allow for greater flexibility, OMAR programs have the ability to take in external data before they run, and then use that data internally. Studio passes these external data into an OMAR program in the form of *program arguments*, which are also sometimes called *command line arguments*. When a Hypercosm applet is included in a web page, its program arguments are given in the HTML code.

An OMAR program reads its program arguments in two phases. In the first phase, the program looks for any set of default arguments. These are arguments that are pre-defined and uniform for all OMAR programs, and they control such things as the graphics window's width and height, its background color, and various other viewing and rendering parameters. In the second reading phase, the program reads any remaining arguments. This second phase is what you program in OMAR code when you use the guidelines set out for using program arguments in the *OMAR Programming Language Reference Manual and Programming Guide*.

For many applications, program arguments can prove to be extremely useful. You could create an applet that reads in varying financial data and then displays those data in 3D. You could create an applet that reads in ticket data and then displays the view from a particular seat in a stadium. The possibilities are almost endless.

To set program arguments in Studio, select **Project > Settings**, and then select the **Applet** tab.


Project Settings

Fixed Size: Width: 768 Height: 640

Scalable Size: % Width: 100 % Height: 100

Arguments:

You can type in whatever set of program arguments you wish in the **Command Line** field. The program arguments you set there are passed to your OMAR program when you run it in Studio, and are also embedded in the HTML file that Studio creates when you build the web page for the project's applet, so that the web applet also receives the arguments.

By default, Studio always passes at least two program arguments to an OMAR program: one that sets the graphics window's width, and another that sets its height. These arguments are read in the first phase of argument reading, and should not interfere with any other arguments that your OMAR program may be expecting in the second phase of reading.

The width and height values you see in the window are the values Studio will place in your HTML file as program arguments when you build a web page. You can change the height and width of the applet your project produces by adjusting these values.

Running & Testing Your Programs

Whenever you want, you can run the program you are working on to see how it looks. When you run a program in Studio, the results display in a Hypercosm graphics window, which pops up as soon as the program is compiled.

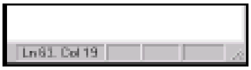
Select **Build** > **Run Projec**t to see what your project looks like. **Run Project** runs the file that is set as your project's main file. The main file doesn't have to be open in order to run it. Another command, **Build** > **Run File**, runs only the file that is currently open and active. This is a good way to test individual files. **Run Project** and **Run File** can also be selected by clicking on the corresponding buttons on the toolbar.

Hypercosm Studio compiles your program and displays the resulting applet in the graphics window. As it compiles, it displays a running commentary in the output window at the bottom of your screen. If the file compiles correctly, the output window tab changes from **Build** to **Debug**; see below for a discussion of the different modes of the output window.

If the program contains errors, you see a message in the output window telling you the line number that contained the error. If you double-click on the line in the output window that tells you where the error was, Studio opens the file and goes to that specific line of code.

Remember that if you click on the elevator button in the scrollbar Studio pops up a box that tells you the number of the line appearing at the top of the window. This is especially handy for large programs.

To determine the current position of your cursor in the file you can use the line and column index located at the bottom right corner of Studio.

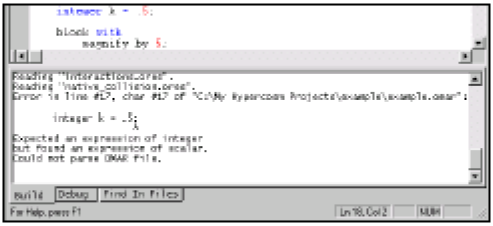


Once your program is running, you can stop it by closing the graphics window, by clicking on the STOP button on the toolbar, or by selecting **Build > Stop**.

If you are already running one program, and you attempt to run another, Studio reminds you that you are already running one, and asks you if you want to kill that one, and run the one you selected. You can disable this feature using the checkbox **Prompt when killing previous renderer** in the window under **View > Options > Misc**. If you disable the message, then Studio automatically kills the previous program before starting up a new one.

Using the Output Window

Hypercosm Studio uses the output window to communicate with you. The output window has three modes: **Build**, **Debug**, and **Find in Files**. Each mode displays a different set of messages for you. You can switch among modes by clicking on the different tabs.



```
    integer k = .5;

    block with
        magnify by 5;
```

```
Reading "interactions.ores".
Reading "native_collisions.ores".
Error in line #17, char #17 of "C:\My Hypercosm Projects\example\example.omar":

        integer k = .5;
                      ^
Expected an expression of integer
but found an expression of scalar.
Could not parse OMAR file.
```

Build mode

In **Build** mode the output window displays messages from Studio's compiler. These messages occur after you compile a file or files, after you build a web page, and after you choose to run a file or files (because Studio must compile the file in memory in order to run it). If the file compiles with no errors, then there are no messages and the output window switches to the **Debug** tab (see below). However, if the compiler does find errors in your file, it stays on the **Build** tab. You can then double click on the error message that contains a line number of an error, and Studio takes you right to that line in the file.

Debug mode

In **Debug** mode the output window displays messages generated by the **Run Project** and **Run File** commands that occur after the file has compiled.

Find in Files mode

In **Find in Files** mode the output window displays the results of a **Find in Files** command. Use this when you are trying to find an OMAR file that contains a specific string of characters. To use this feature, select **Edit > Find in Files**. Then type in the string of characters you are searching for. Studio displays the results of the search in the output window. If you double click on the search results in the output window, Studio opens the file and jumps to the specific line number.

You can specify whether you want the output window to clear before each program runs; the window under **View > Options > Misc**. provides a checkbox **Clear output window before running**. On this window you can also control whether the output window is on or off when a program runs using the check-box **Show output window when a program is run**.

Adding an Applet to a Web Page

Once you've created an OMAR program and an accompanying Studio project, you can then create an applet and add it to your HTML file. What you should read now depends on how you want to do this:

- If you want to test your applet by adding it to the template HTML file that was created when you started your project, you can skip ahead to the "Building the Web Page Automatically" section.
- If you want Hypercosm Studio to insert your applet *automatically* into an existing web page, then read the "Preparing Your HTML File" section, followed by the "Building the Web Page Automatically" section.
- If you want Studio to generate code that you can then insert by hand into an existing HTML file, then read the "Inserting Hypercosm Code by Hand" section.

  Note: to insert two *different* applets into the same page, you have to insert them by hand because Studio cannot insert two different applets automatically. Also, if you want to insert an applet that creates multiple windows, you'll have to make the appropriate changes to your HTML file because Studio cannot yet do so.

Preparing Your HTML File

Hypercosm Studio can insert your applet into your HTML page automatically. Remember that Studio first makes a copy of your HTML page, and inserts the applet into the *copy*, not into the original page.

To do this, you must first do two things:

- Hypercosm Studio must know where to find the HTML file. If you haven't already added the HTML file to your project, follow the directions given in the "Adding Files to Projects" section.
- Hypercosm Studio must know where *in* the HTML file to add the code. The rest of this section explains how to specify the applet's location in the your HTML file.

Follow these directions to specify the location of the Hypercosm applet in your HTML file.

1. Open the HTML file with a text editor or with your web design program.
2. Move to the spot in the HTML file where you want to call the Hypercosm applet.
3. Insert the following two lines of text at exactly the spot where you want the applet to appear. These lines are case sensitive, so be sure to type them exactly as they appear. If you place any text between the two lines, Studio will delete it when it inserts the applet.

   <!--HypercosmAppletBegin-->

   <!--HypercosmAppletEnd-->

Note that if you have coded your HTML file by hand, you can just type in the text as it appears here. If you are using a web design program, you must make sure that your program won't modify this text in any way, either because it doesn't understand it, or it thinks it's unnecessary. Most web design programs give you the capability to add comments or code that is used by other applications; check the documentation that came with your web design program if you are unsure how to do this.

Once you've inserted this text into your HTML file, and saved the file, you are ready to have Studio build the web page automatically.

Building the Web Page Automatically

When Hypercosm Studio builds a web page, here's what happens:

- Studio makes a copy of the project's HTML file, and places that copy in the subdirectory called *project***Website**\ where *project* represents the name of your project. Studio also copies any necessary sound or texture files to the **Website** subdirectory. If you'd prefer that all of these files be placed in a different directory, you can change the default "Web Output Directory" in your project's settings.
- Studio compiles your program into HCVM format (Hypercosm's byte code format), and places the resulting HCVM file in your project's directory (or wherever you specified the "output file" be placed in your project's settings). Studio then copies that HCVM file and places the copy into the "Web Output Directory" (which is the **Website** subdirectory by default). Note that this effectively creates *two* HCVM files: one in your project directory, and one in your web output directory.
- Studio opens the copy of the HTML file and looks for the string that tells it where to insert the applet. The template HTML file that was created when you created the project already contains this string. If you are working with your own HTML, then you should have already inserted this string yourself, using the directions given in the "Preparing Your HTML File" section. Once it finds the string, it inserts a reference to the HCVM file (known as an "applet tag") into the copy of your HTML file.

To build a web page, choose **Build > Build Web Page > Scaled** or **Fixed Sized**, or click on the **Build Web Page** button on the toolbar. The settings for **Scaled** or **Fixed Sized** can be set by going to **Project** > **Settings** > **Applet.** The Fixed Size setting will place the applet in your html file that will have a width and height of specified pixel values. The Scaled setting will place the applet in the html file according to a percentage of the total html page size.

You can follow the progress of the build in the console. Studio displays any messages there. Once the process is complete, Studio displays a success message in the output window.

Inserting Hypercosm Code by Hand

Some users prefer not to have anyone or anything modify their HTML files. If you are this sort of user, Hypercosm Studio can accommodate you. Studio can compile your applet into HCVM format (Hypercosm's byte code format). You can then copy an applet tag into your HTML file at the point where you want to call the Hypercosm applet. Note that you must use this method to include different Hypercosm applets in a single web page.

Briefly, what you need to do is build a web page using the template HTML file that was created when you created your project. Use the directions given in the "Building the Web Page Automatically" section. The process of building the web page automatically compiles your applet into HCVM format, and creates the HTML code necessary to call the applet.

Once you've created the sample web page, open the output HTML file (in the **Website** subdirectory) and copy everything between the tags

<!--HypercosmAppletBegin--> and **<!--HypercosmAppletEnd-->**

to the clipboard. Then open your HTML file and paste the clipboard contents into your HTML file at the point where you want to call your Hypercosm applet. You must also move any sound or texture files your applet uses into the directory where your HTML file is located.

Viewing the Web Page

If you have built your web page automatically, then all the files required to view your web page should be in your project's **Website** subdirectory (or in whatever directory you set as the web output directory in your project's settings). In this case, you can use Studio to see your web page after you've built it. Simply select **Build** > **View Web Page**.

If the appropriate files are located in the **Website** subdirectory, Studio launches your web browser and opens your web page, complete with the Hypercosm applet. This will not work, however, if you have inserted the code by hand, and your web page is not located in the **Website** subdirectory.

Note that you can specify which web browser Studio should launch in the window under **View** > **Options** > **Web Browser**.

Compiling a Project into an Applet

Once you've added an applet to a web page (using one of the methods described above), it is unnecessary to edit that page's HTML file every time you make changes to the applet it contains. If you change some operability in the applet itself, you need only compile your revised OMAR code into an applet, replace the old applet, and then when you open the previously built web page, the new, revised applet should appear.

To compile a project into an applet, without building a web page, select **Build > Compile Project**. The newly compiled applet is placed in your project's main directory. If you want it to appear in your project's web page, you'll have to move the applet to the directory where your HTML file is located (which is the **Website** subdirectory by default).

Studio also allows you to compile a single OMAR file by selecting **Build > Compile File**. Note that Studio can only build a web page for a complete project, not for a single file, so you should generally only produce your applets from complete projects rather than from single OMAR files.

There are a few cases in which changes to an applet require changes to the HTML file that includes it. Any time you change an applet by including new sound or texture files in it, you have to add those files to your project, and you also need to build your web page again, so that those new files are included in the HTML code. You must also rebuild your web page when you change any program arguments that are passed into your applet, including the width and height arguments.

Creating Your Final Product

Once you've created an applet, you must copy all necessary Hypercosm files to your web server. These files include your HTML file, the HCVM file that contains the compiled OMAR code, and any texture or sound files needed by the applet. If you've had Studio build your web page automatically, you can find all of these files in the **Website** directory your project created, or in whatever directory you specified as your project's Web Output Directory in the Project Settings.

Note that, no matter what changes you might make to the HTML that Studio generated, your HCVM file must be placed in the *same directory* as the HTML file that includes it for the applet to appear properly in the web page.

Downloading the Hypercosm 3D Player

Through the Hypercosm web site (http://www.hypercosm.com), you can download the Hypercosm Player (http://hypercosm.com/download/player/index.html) to view your applets. If a user is trying to view an applet without the Hypercosm 3D Player installed the applet will automatically redirect the user to the download page.

Communicating with Hypercosm LLC

If you should have any significant problems with your Studio product, or if you have specific requests for new features, you can connect to the Hypercosm web site and report your concerns.

To communicate with Hypercosm LLC, use Hypercosm's Help web page (http://hypercosm.com/support/index.html). Choose **Submit Report Bug** to write to us about a potential program error, and choose **Submit Feature Request** to request new product features.

Hypercosm News, Support, & User Forums

We recommend that frequent users of Hypercosm products also be frequent visitors to our web site (http://www.hypercosm.com). There you can find technical support contact information, OMAR development tips and FAQs, and news of all our latest updates and releases. You can also read or contribute to Hypercosm's User Forums, which will keep you informed of all the latest Hypercosm news, and allow you to communicate with other Hypercosm users. In fact, if there's something you'd like to see on our web site that's not there, let us know, and we'll do our best to provide the service you request.

Hypercosm™

We Put 3D To Work!℠