

# MAX



## What's New in Max 4

# Introduction

This manual is for users of version 3.x of Max. It highlights some significant new features and explains important differences between Max 4 and previous versions. There are three chapters that follow: *What's Been Added*, *What's Been Deleted*, and *What's Been Changed*.

## Installing

Keep your Max 4 installation separate from any previous versions of Max. Most of the new and changed objects do not work with previous versions, and you will create a mess if you try to combine old and new.

You can, however, drag third-party external objects into your Max 4 folder. Most third-party objects work with Max 4.

You won't be able to use any Max 4 external objects with previous versions. They will not be recognized.

## Copy Protection

Your existing Max authorization will not authorize Max 4. However, you have 30 days to obtain a Max 4 authorization (using challenge/ response: for more details on this authorization refer to the **Read Me Before Installing** Document in the folder containing the Max installer).

There are two kinds of authorizations, one for Max 4 alone and one for Max/MSP. There is no separate MSP authorization.

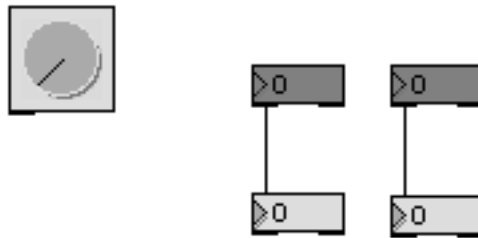
## MSP and Max

The use of MSP should be mostly transparent, since there is a single Max/MSP installer and the Max Audio Library is built into the Max application now. This document describes only those changes made to the Max environment and its objects. Please see the **What's New in MSP 2** documentation for details about changes to MSP.

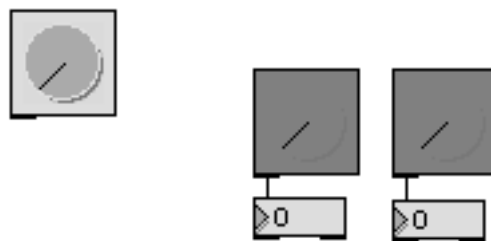
# What's Been Added

## Edit Menu

In addition to the new behavior of Max's find and replace feature covered below under the Find Window, the Edit Menu features a powerful new editing command called **Paste Replace**. There are a number of uses for Paste Replace; the main one is that you often change the settings of a single object (such as a **number box**) and wish to apply them to several objects. Previously, the only way to do this was to duplicate the changed object, delete the unchanged ones, and manually recreate the connections you deleted using the duplicates. Paste Replace can perform this operation with far less effort. To use Paste Replace, you copy a single object, select one or more objects, then choose **Paste Replace** from the Edit menu. The copied object will replace all the selected objects, and the connections to the selected objects will be restored (to the extent possible given the pasted object's inlets and outlets). Paste Replace is only available when a single object has been copied to the clipboard and one or more objects have been selected.



*Before Paste Replace (with a dial Object on the clipboard)*



*After Paste Replace*

# What's Been Added

---

## View Menu

The all-new View menu controls what you see when you are editing a patcher window. The top item in the menu is **Edit** (formerly in the Max menu), which locks and unlocks a patcher. Here's a brief description of the other items.



### Hide Object Palette / Show Object Palette

Controls whether you see the palette of icons for user interface objects at the top of the patcher window. With this invisible, you can use the new contextual menu feature of a patcher window to make a new object (see below). Hiding the object palette makes it easier to adjust the appearance of the top part of the patcher window.

### Hide Connections / Show Connections

Controls whether all patch cords are hidden when the patcher is unlocked. Hiding patch cords makes it easier to adjust the positions of objects in a patcher window; since clicking on a patch cord is prioritized over clicking on a box, if you have a rats nest of patch cords over a box, you'll have a hard time clicking on it.

When the patcher is locked, only those patch cords that are designated Hide on Lock are hidden, regardless of the setting of Hide Connections.

### Hide Foreground / Show Foreground

Hides the objects that belong to the foreground layer, making it easier to adjust the background layer. By default all objects belong to the foreground layer (including all objects in existing Max patcher files) unless you move them to the background by choosing **Include in Background** from the Object menu when an object is selected. Background objects always draw behind foreground objects.

### Hide Background / Show Background

Hides the objects that belong to the background layer, making it easier to adjust the foreground layer. By default all objects belong to the foreground layer (including all objects in existing Max patcher files) unless you move them to the background by choosing **Include in Background** from

# What's Been Added

---

the Object menu when an object is selected. Background objects always draw behind foreground objects.

## Lock Background

When this item is checked, objects in the background layer do not respond to clicks when the patcher is unlocked. This allows you to edit objects in the foreground layer on top of them as if they were “wallpaper.”

## Set Origin / Restore Origin

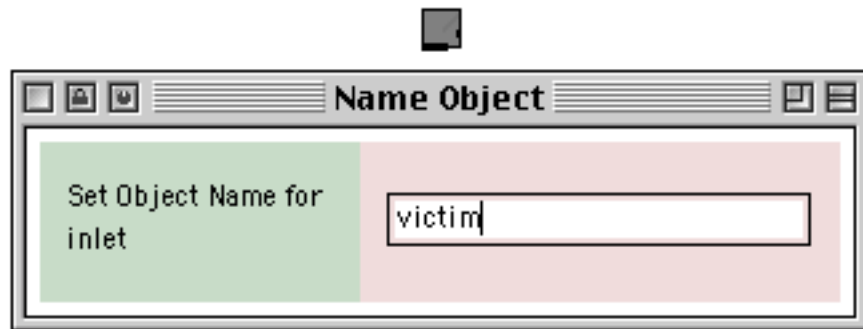
These items are enabled only if you have scrolled a patcher window so that its original (or set) top-left corner has changed. Choose **Set Origin** to save the current top-left corner as the origin. This location will be saved with the patcher and always appear as the top-left corner when the patcher is opened. Choose **Restore Origin** when you have scrolled the window away from the origin you set and want to scroll back to it quickly.

## Object Menu

The Object menu, formerly the Max menu, has three new items and one that is radically improved, so we'll mention it here.

### Name...

This item opens a window for giving an object a variable name. Variable names are exclusively for the new scripting features of the **thispatcher** object; they currently have no other function. See the new Max tutorials on scripting (Tutorial 46 and Tutorial 47) as well as the reference pages for the **thispatcher** object for more information.

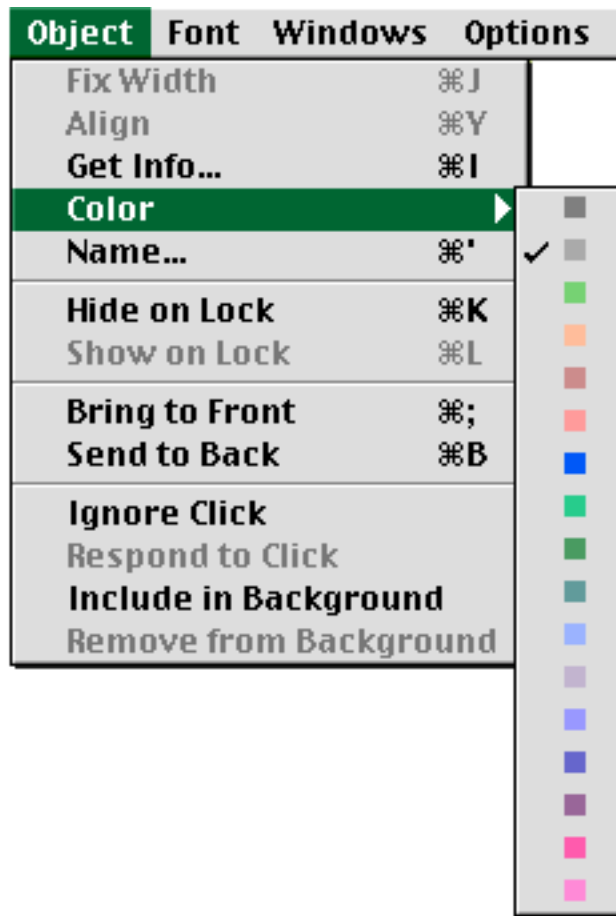


### Color

This item formerly opened a dialog box. Now it is a submenu that lets you change an object's color in one step. You can change the colors displayed in the menu (which are, by the way, global to all

# What's Been Added

patchers) by choosing **Colors...** from the Options menu. Another thing about the Color submenu is that you can now change the color of patch cords in addition to objects.



Many objects allow you to change their colors with RGB values; this is normally done using the object's Inspector window.

## Get Info...

This item no longer opens a dialog box to set properties of the standard Max objects. Instead, an Inspector window is shown. For more information about Inspectors see the Inspector section below.

## Include In Background

If one or more objects are selected and are currently part of the foreground layer, Include in Background moves them to the background layer.

## Remove From Background

If one or more objects are selected and are currently part of the background layer, **Remove From Background** moves them to the foreground layer.

# What's Been Added

---

## Windows Menu

### Show Floating Inspector / Hide Floating Inspector

Opens a floating window that displays the Inspector for any object that is selected. See the Inspector section below for more information about this new feature. This feature requires Mac OS 8.6 or later. If Max is running on an earlier system, this menu item will not be present in the Windows menu.

## Options Menu

There are a number of new items in the Options menu.

### Float Display Correction

When this option is checked, floating-point numbers are rounded to what the program believes is an appropriate number of digits when they are displayed. So you would see 4.0 instead of 3.99998. This in no way affects the computation performed with floats in Max.

### Help in Locked Patcher

In Max 4, option-clicking on an object in a locked patcher no longer opens its help patch, unless this option is checked.

### Enhanced File Preview

Max's standard open file dialog includes a preview feature, mainly used for audio files and QuickTime movies. If Enhanced File Preview is on, the preview dialog will attempt to use QuickTime to preview files such as text documents and graphics files even if they do not have preview information stored in them. If this is not checked, previewing only happens for standard types of files. Turn this option off you find the open file dialog is too unresponsive.

### DSP Status...

When using Max/MSP, this item is enabled and opens the DSP Status window.

### Colors...

Opens a window where you can edit the standard colors used by a number of objects (and now, patch cords); your changes are stored in Max's preferences file.

## Extras menu

This is a handy new menu that lists all the patches in the extras folder found inside the patches folder. Choose *Adding Items to this Menu* for directions on how to make your own Extras. The Tips patcher contains some shortcuts for editing in a patcher window. You probably don't know all of them, so check it out!

# What's Been Added

---

## Contextual Menus

Contextual menus are available in an unlocked patcher when you hold down the control key and click on something. The exact menu you get depends on what you click on.

If you control-click on an object, you'll get a menu that lets you perform editing operations (such as Copy and Clear) or modify the object's properties. Most of these items are found in the Object menu (formerly the Max menu).

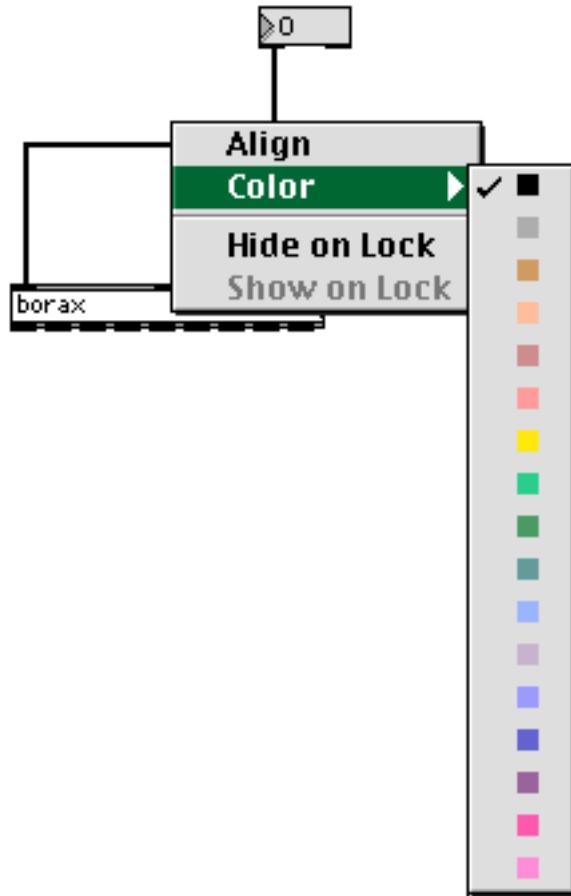




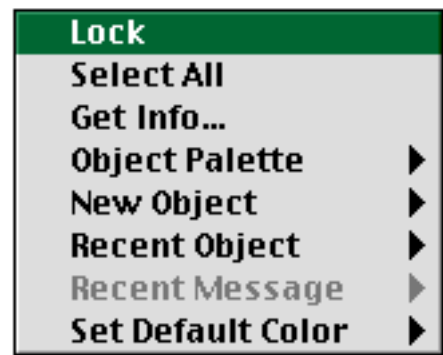
# What's Been Added

---

If you control-click on a patch cord, you'll be able to make it hidden or visible when the patcher is locked, straighten it, and change its color.



However, many new features are available when you control-click on empty space in the patcher.



Here is an introduction to the new items in this menu:

# What's Been Added

---

## Object Palette

Provides you with a submenu showing the palette of available user interface object icons. Choosing an icon turns the cursor into the icon. The next time you click, you'll make an object of the chosen type.

## New Object

Provides you with submenus listing all of the Max search path, plus a list of objects built-into Max. Choosing a name from the menu creates an object box and places the name into the box. Clicking outside the box will create the object (either a standard Max object or your object). Note however that since some valid Max files are stored in text format, all text files found in the search path are listed in this menu, whether they are valid Max documents or not.

## Recent Object

Provides a submenu with the names you have typed into an object box since you launched Max.

## Recent Message

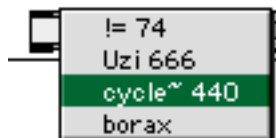
Provides a submenu with the text items you have typed into a message box since you launched Max.

## Set Default Color

Choose a color for from the submenu to use when making a new object or patch cord, rather than the default black/gray.

## Other Contextual Menus

If you control-click on the object box icon in the object palette, you'll get the Recent Object menu described above.



If you control-click on the message box icon in the object palette, you'll get the Recent Message menu described above.



## Inspectors

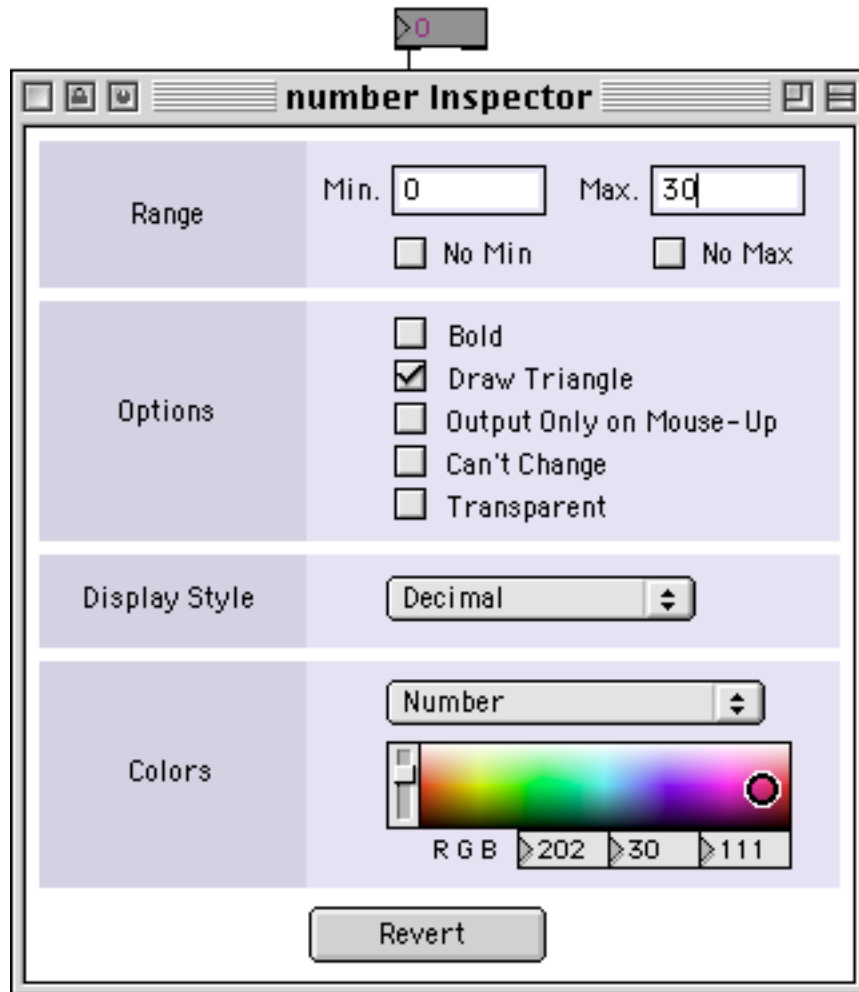
Almost all dialog boxes in Max 4 have been replaced by special patchers called *Inspectors*. These patchers are opened when you choose **Get Info...** from the Object menu with a single object

# What's Been Added

selected. As with previous versions of Max, you cannot inspect more than one object at a time. However, the new **Paste Replace** feature described above lets you apply changes you make to one object to a series of objects.

Changes you make to an object in an inspector occur immediately, unlike the old Get Info dialog boxes, where the changes weren't registered until you closed the window. If the change you make requires the object to redraw, it will redraw in an unselected fashion, which may be slightly confusing. Rest assured the object is still being affected by changes you make in the Inspector window. Indeed, an object need not be selected to be affected by its Inspector. The window will stay open until you close it (or delete the object you're editing).

A feature shared by all inspectors include the ability to revert changes to the state the object was in when you opened the inspector. You can do this in two ways, either by choosing **Undo Inspector Changes** from the Edit menu when an inspector window is frontmost, or by clicking the *Revert* button at the bottom of the inspector window. Another feature shared by most inspectors is the ability to tab from one text field to another (as you would expect in a dialog box). The text fields are implemented using the new `textedit` object described briefly below.



*The Inspector for the number box object, which includes text fields and the Revert button*

# What's Been Added

---

The Floating Inspector window (on older system versions it may be called the Global Inspector) displays the inspector for the currently selected (single) object in a floating window. This feature may take some getting used to, but it can be convenient for getting detailed information about a number of objects very quickly. Unlike normal inspector windows, menu commands do not operate on the floating inspector so you cannot copy and paste text from inspector text fields.

## New Max Objects

There are over 40 new objects in Max 4. Here are some highlights:

You'll be able to create compelling and detailed user interfaces with the nine new Max user interface objects. **pictctrl** can be a button or a dial that uses a file of images you prepare to represent its various states. **pictslider** is similar, but acts as a one- or two-dimensional slider. **hint** displays text underneath its rectangle when the mouse moves over the rectangle. **dropfile** defines a part of a window as a receptacle for file icons dragged from the Finder or another application. **matrixctrl** is a two-dimensional matrix display with customizable appearance using picture files. **panel** is a colored background rectangle that works well with the new background layer feature of patcher windows. **radiogroup** is a set of radio buttons or check boxes used in many of the new Inspector windows. **swatch** is a Photoshop-like color display that produces a list of RGB values. And the **textedit** object lets you enter text into a patcher, something you've probably wanted to do for a long time. You can even use Max menu commands on **textedit** objects when they are highlighted in a locked patcher.

Learn about these objects by investigating their Inspectors, help files, and, of course, documentation in the **Max Reference** manual.

Non-UI objects include:

- a full complement of mathematical operators only available previously (if at all) as functions within the **expr** object including **cos**, **sin**, **cosh**, **sinh**, **pow**, **tan**, **atan2**.
- objects for managing files and pathnames: **opendialog**, **savedialog**, **filepath**, **strippath**, **relativepath**, and **absolutePath**.
- objects for symbol and list handling: **zl**, **tosymbol**, **fromsymbol**, and **substitute**
- the **appledvd** object for controlling the Apple DVD Player application using Apple Events
- objects for getting information about the current execution environment: **suspend**, **gestalt**, and **screensize**.

Many existing Max objects have been significantly enhanced, see the Changed Max Objects section for highlights.

## Editing and Display Enhancements to the Patcher Window

The patcher window has a number of new features that support the construction of complex layered interfaces. A number of the new Max objects such as **panel**, **pictctrl**, and **radiogroup**, are designed to take advantage of these features. You can for example, put a **comment** object over a pic-

# What's Been Added

---

ture or colored background without seeing the white rectangle surrounding the text (you will see the white rectangle when the patcher is unlocked however). You can also put a **radiogroup** object in front of a background and you'll be able to see what's between the individual buttons or check-boxes. Or notice how the background bleeds through to the corners of the newly enhanced **umenu** object. These are subtle features but they let you move your Max interfaces away from the strictly rectangular affairs you were forced to use in the past.



Other enhancements include:

- Changing an object's rectangle by dragging on its grow box displays the current size in pixels in the patcher's assistance area
- Option-dragging on an object creates a copy you can move to where you want, just like a paint program. Shift-option-drag constrains the item to be in line with the original (although it may not be perfectly lined up).
- To make more than one instance of a user interface object, shift-click when the cursor has changed to the icon of the object. The cursor will stay the same, so you can click (or shift-click) somewhere else to make another one.
- You can type the name of a user interface object into an object box, and the box will change into the interface object.
- You can type the name of an object into an object box without worrying about how it is capitalized (or not). For instance, you can type "uzi" and obtain the **Uzi** object.



*These two objects are the same*

# What's Been Added

---

## Scripting

The **thispatcher** object accepts a new set of messages that begin with `script`. These allow you to create, connect, disconnect, and move objects in a patcher as well as change object properties and send objects messages without patch cords. Scripting is not yet a full-featured language for controlling Max, but you can do some fairly powerful things. There are two new tutorials (Tutorials 46 and 47) that cover some scripting concepts, and a new section of the reference pages for the **thispatcher** object that describe each of the new scripting commands.

# What's Been Changed

## Menus

The File menu has a New submenu that replaces the New menu.

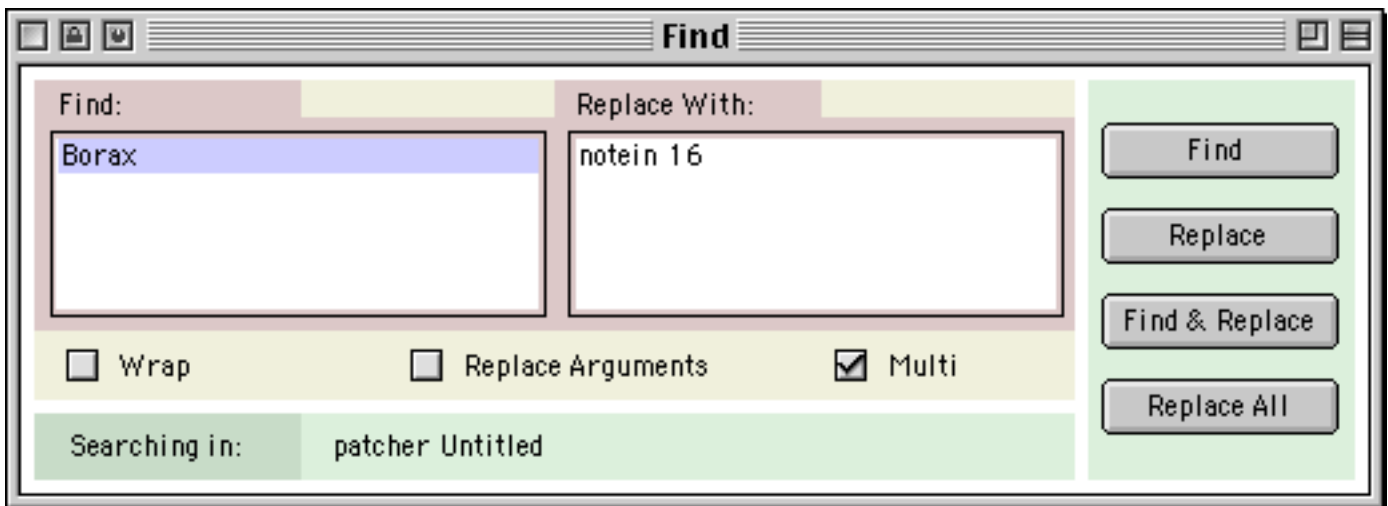
The Max menu has been renamed the Object menu.

The **File Preferences...** item has been moved to the Options menu.

The **Edit** item in the Max menu has been moved to the View menu.

## Find Window

The Find window is now a patcher, and in addition to the enhanced appearance of the interface and larger text entry areas, it supports some new features such as searching in multiple patchers (check the Multi option).

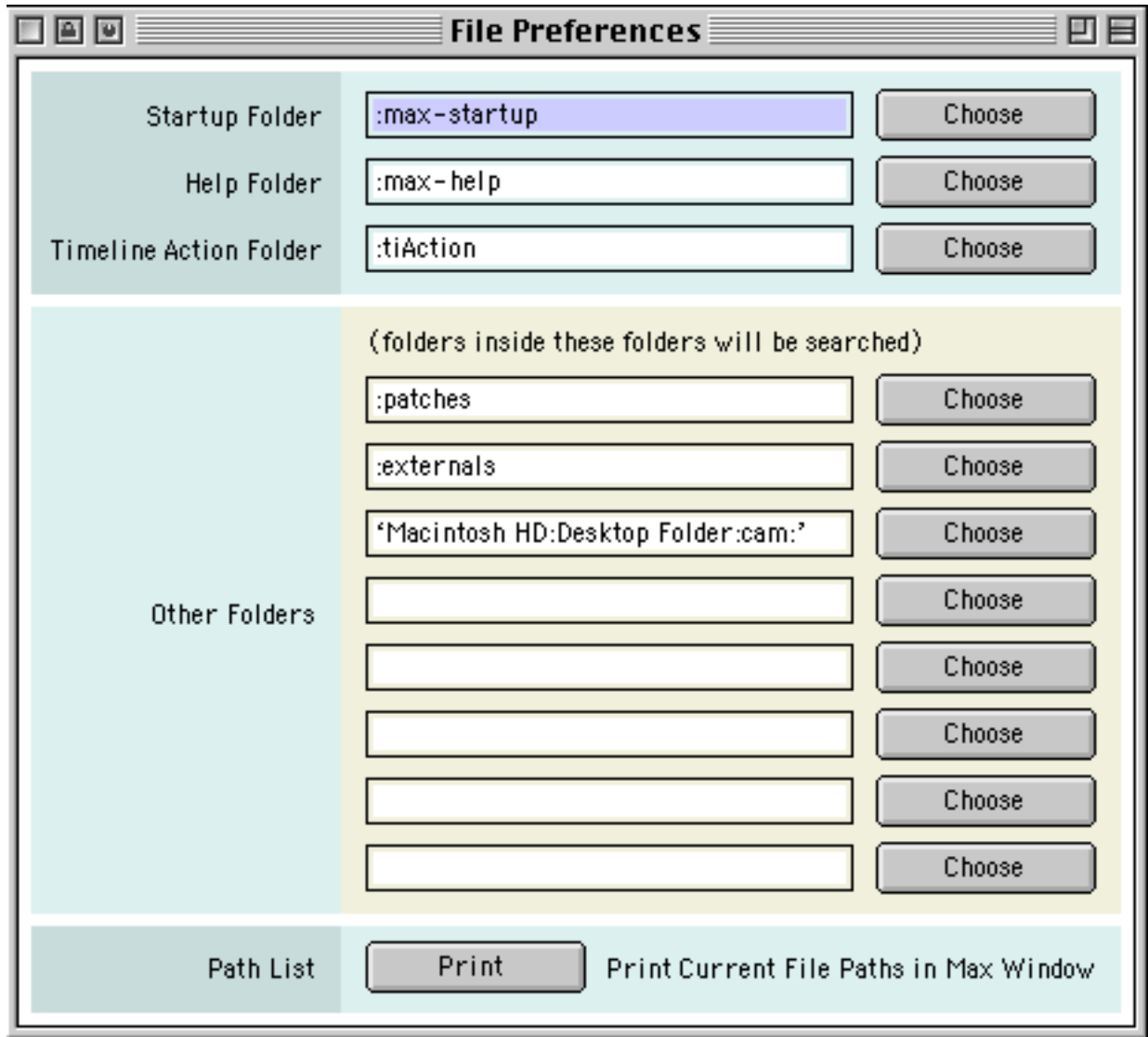


## File Preferences

The File Preferences dialog has been changed to a patcher window, and contains a number of enhancements. There are now two additional slot for folders in the search path (and if you wanted, you could store up to 256 search path locations, consult the documentation on the `filepath` object for details). In addition, you can use the standard folder choosing dialog to specify search path locations (click the *Choose* button next to any slot). To verify that your search path is what you

# What's Been Changed

want, click *Print* at the bottom of the dialog. This will list all of the folders being searched in the Max window.



## Text window

The text editing window (obtained by opening a text file that isn't a patcher, opening a patcher using **Open As Text...**, choosing Text from the New submenu of the File menu etc.) now has undo and is not limited to displaying 32,000 or fewer characters.



# What's Been Changed

---

## Patcher-related changes

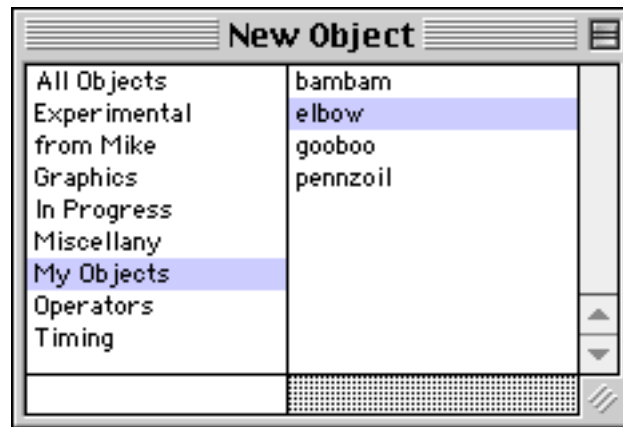
### Disabling Loadbang While Loading Patches

In previous versions of Max, holding down the command key while loading a patcher temporarily disabled all **loadbang** objects. In Max 4, you need to hold down both the command and shift keys in order to disable **loadbang**. This change was made because we found during testing that users were opening Inspector windows using the command-I shortcut, which defeated the critical **loadbang** operation the inspectors used to display the current state of an object.

### New Object List window

When **New Object List** is checked in the Options menu, the New Object window appears when you create an object box. A number of minor enhancements have been made to this window that let you customize it to create a personal catalog of objects.

First of all, there are no longer any fixed categories for Max objects, so if you don't like the ones we supply, you can simply edit the Max Object List file (found in the same folder as the Max application) and rename the categories. Second, you can add new categories (up to 100) simply by listing them in the Max Object List file. Max/MSP users will see that there are quite a few MSP categories displayed in the New Object List. The only category you'll have to live with is All Objects, which simply lists every object added to the list.



*Example of a customized New Object List*

Finally, you can change the size of the New Object List and the size is stored as a preference so the window will be sized properly the next time you launch Max.

### Obtaining a list of messages to an object

Previously, shift-option-clicking on an object displayed a menu that listed all the messages you could send to that object. This feature has been changed to control-option-clicking, since shift-option-clicking is now used with the new drag-duplicate feature. The change makes logical sense too, since control is used for contextual menus, and option is used for getting help on an object.

# What's Been Changed

---

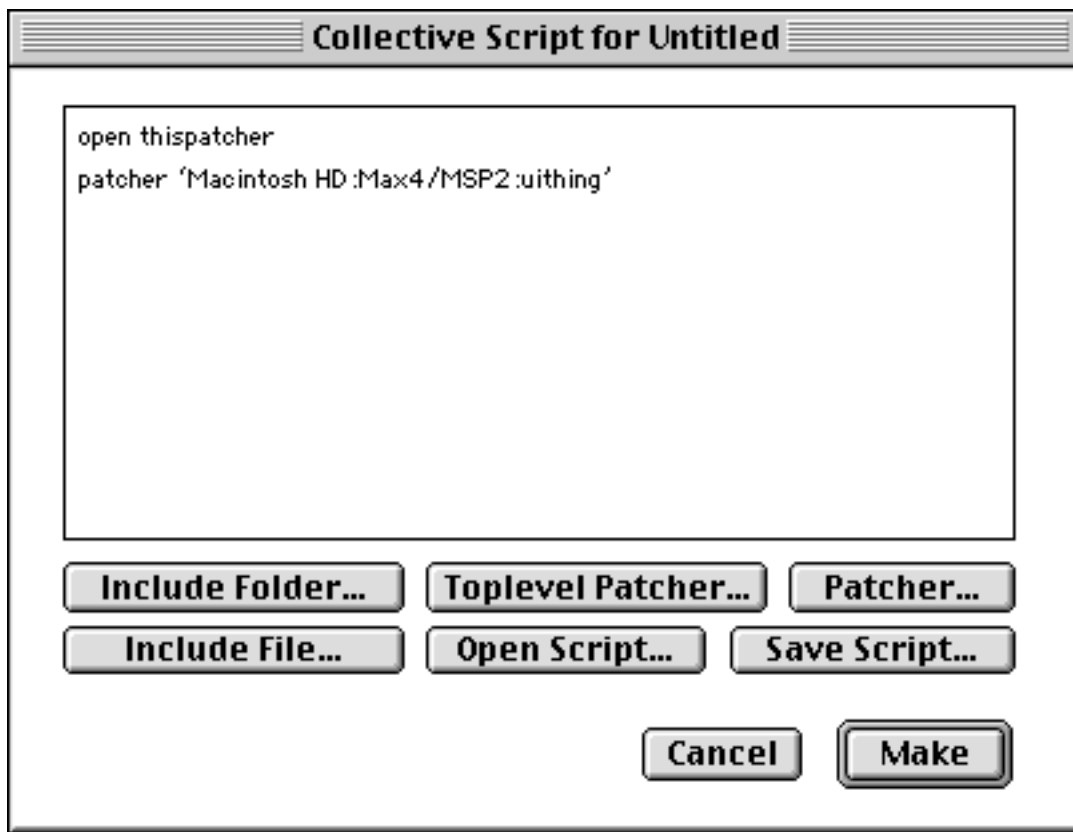
In addition, the message list menu contains two items you can actually choose at the very bottom. **Open <objectname>.help** opens the object's help file (if available), and **About This Menu...** opens a patcher called *boxquickref.help* that explains the message list syntax.

## Paste in a locked patcher

In previous versions, **Paste** in the Edit menu would be enabled if a patcher were locked but there were objects on the clipboard to paste. Choosing **Paste** would unlock the patcher and paste the objects. In Max 4, the Edit menu in a locked patcher can operate on objects that handle copying and pasting of data (such as the *textedit* object). Therefore, you can no longer use Paste in this way, and you'll need to unlock the patcher before pasting objects.

## Build Collective Dialog

A new *Patcher...* button lets you choose a patcher file and, when building the collective, will analyze the patcher file for its dependencies and include any externals or other files needed by the patcher. It uses the patcher keyword in the collective script, which was an undocumented feature of previous versions.



The *Include File...* feature has been enhanced to let you open any file that you would like to include in a collective.

The collective dialog window is now moveable.

# What's Been Changed

---

## MIDI Setup dialog and OMS MIDI Device Select dialog

These dialog windows are now moveable.

## Open and Save Dialogs

On newer systems (OS 8.6 or later), the standard open file, choose folder, and save file dialogs use Apple's Navigation Services. An optional preview feature lets you preview a variety of files using QuickTime.

## Save Changes Dialog

You can type cmd-D to cancel out of the dialog (D as in **D**on't Save).

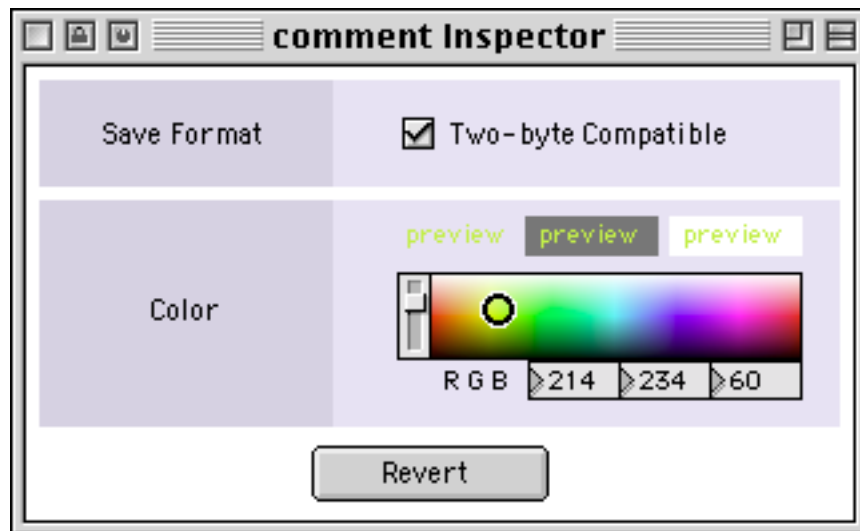
## Significant Changes and Enhancements to Max Objects

### External Object Files

The "file type" of external objects was changed so that their icons show up properly in OS 9. This means that you won't be able to load any Max 4 objects into previous versions of Max.

### User Interface Objects

The **comment** object can be colorized, both using the standard colors in the Color submenu of the Object menu as well as with RGB values in its Inspector. It no longer draws a white rectangle around its text in a locked patcher. You can also use the Inspector to set a two-byte-compatible mode (previously available by using the **com** object) that is recommended for saving text on Japanese systems and others that employ two-byte characters.



The **number box** features an additional outlet that reports when a tab key has been pressed when it is the highlighted object (in other words, the object that receives keyboard input). You can use this to implement an interface in which the user can tab from one number box or textedit object to the

# What's Been Changed

---

next. Also new are a transparent mode that can be set in the Inspector, as well as customization of text and background colors. comment objects do not appear in color until you lock the patcher.

The **bpatcher** file dialog has been replaced with an Inspector, which, unlike the previous version, does not appear the moment you create a new **bpatcher**. To choose a file to display inside a **bpatcher**, click the *Choose* button in the inspector, type a filename into the text entry area, or drag a file icon onto the text entry area. Note that, as has always been the case, selecting a file for a **bpatcher** that is outside Max's search path is problematic. Since the name is stored as an absolute reference, if you move the file or rename a directory, the file could be impossible for **bpatcher** to find.

A number of enhancements have been made to the **fpic** object, allowing you to use powerful QuickTime image conversion and scaling features when opening a file to display.

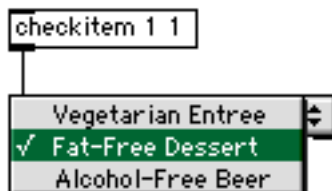
The **imovie** object has a number of powerful new features. See the discussion of the **movie** object below for a summary.

The **kslider** object sports a revised customizable appearance and the ability to “play” more than one key at a time.

The **lcd** object has been completely rewritten. It features offscreen, sprite, region, and clipping support, the ability to read pictures, and many other new features.

The **ubutton** (transparent button) object has additional outlets to report positional information. You can also obtain the cursor position while the user drags on the button by checking *Track Mouse While Dragging* in the object's Inspector.

The **umenu** object no longer opens a dialog box when you create a new instance. Choose **Get Info...** from the Object menu to use the Inspector to add items. The appearance of the object has been enhanced and can be customized with your own colors in the Inspector. There are also new messages for checking individual items, and the object is transparent when you use it in Label Mode.



## Other Objects

Many standard Max objects handle floats, lists and symbols where before they dealt exclusively with ints. Here is a partial list:

- The **route** and **select** objects now accept arguments that are a mix of ints, floats, and symbols; there is no longer a limit to the number of arguments you can type in.

# What's Been Changed

---

- The **split** object accepts floats as arguments
- The **bondo** object can synchronize lists and other messages in addition to ints
- The **change** object works with floats if you supply it with a float argument
- All timing objects accept float arguments and messages (see Floating-Point Scheduling below)
- The **match** object can now match floats, symbols, messages
- The **line** object accepts a float argument that makes it output float values. Whether in int or float mode, it accepts floats to specify times in either int or float output mode.
- **iter**, **thresh**, and **cycle** accept floats

## Improvements to the movie and imovie Objects

- Rates can now be specified in floating-point
- A powerful set of matrix messages for image manipulation
- The read message to the objects opens the same set of 25 file types that the Quicktime player opens. In addition, there is a **readany** message that will open any file type. This means you can play MP3 files etc. in a movie object without having to convert the file type to the standard movie type (MooV).
- You can select framed or frameless appearance for imovie, and choose a window with a one-line border for movie
- You can loop over an arbitrary section of a movie
- With the **quality** message, you can control the amount of time spent playing the movie, which allows you to allocate more time to hard-disk audio playback or other tasks

## File Handling

- Most objects that accept read and write messages now let you supply arguments to these messages to
- Objects that accept read messages with an optional argument now let you supply a full or partial pathname to specify a file location. This feature was unintentionally supported in previous versions. A full pathname begins with the name of your hard disk, and a partial one is relative to the folder containing the Max application. For instance, a file in the patches folder could be specified as `:patches:myfile`.

## Random Algorithms

- You have greater control over the random number generation in the **random** and **drunk** objects. These objects accept an optional argument that initializes their seed value (or you can send

# What's Been Changed

---

them a seed message to do it). If you set two different objects to have the same seed value when they are created, they will output an identical sequence of values. The default seed is 0, which uses the system timer to generate a seed value. The value is now guaranteed to be unique for all instances of a particular class. In other words, by default, all **random** objects will output unique sequences of numbers.

## Miscellaneous Changes to Objects

- The **gate** and **switch** objects accept a second argument that sets their initial value. The number of inlets or outlets to these objects is now 100 rather than 10.
- The **trigger** object allows you to output “literal” values in addition to conversions of the input. For example `trigger i 20.5` will output 20.5 followed by the input as an int. Previously you would have to do this sort of thing using a `b` argument to `trigger` followed by a message box containing 20.5.
- The **cd** object has a new `trackdurs` message that prints out the duration of each track in milliseconds.
- The **key** and **keyup** objects have an additional outlet that reports the states of the modifier keys (shift, control, option, etc.) when the key was pressed.
- The **coll** object has an additional outlet that sends a bang after a dump is completed.

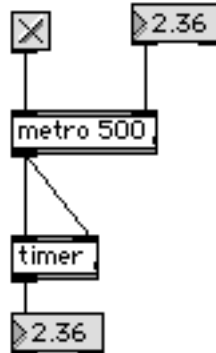
## Floating-Point Scheduling

The Max scheduler now operates in double-precision floating point. Since it only runs every millisecond, what good is this? There are two reasons. First, by using double-precision to represent milliseconds, we've dramatically increased the amount of time you can run a patch without the internal time value of the scheduler wrapping around to 0 from a couple of days to several decades. Second, when using Max/MSP and the Scheduler in Audio Interrupt feature, the scheduler will run every signal vector size. This can be as low as two samples (effectively about 20,000 times a second). While using this particular value dramatically lowers the amount of processing you can do, it means that you can almost do sample-level signal processing entirely in Max! More important benefits are dramatically increased accuracy in timing of audio-based changes: a **metro** can be set to 0.5 milliseconds, and, with a signal vector size of 16, the scheduler will actually run more often than this (about 0.36 ms), so you'll actually get something close to events this often. Of course, running the scheduler this often to generate MIDI information isn't worth doing, since the timing resolution of MIDI is well above this amount of time.

# What's Been Changed

---

Objects affected by the change to floating-point scheduling include **metro**, **clocker**, **pipe**, **delay**, **tempo**, **setclock**, **line**, and **timer**. The objects will work identically to previous versions when you supply them with int arguments.



*Logical time is preserved independent of how often the scheduler is called*

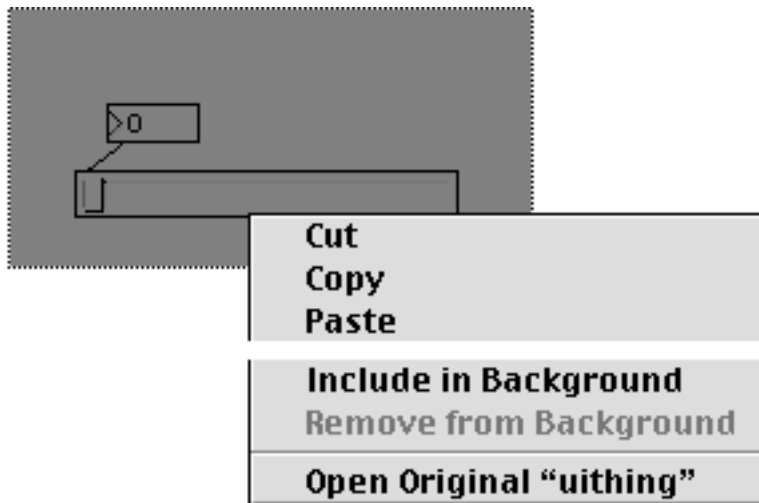
# What's Been Deleted

The **Configuration...** dialog has been removed, since the vast majority of computers running Max are now powerful enough to run the scheduler at 1ms intervals. If you still want to adjust the scheduler interval, you can do so by sending a message to the Max object. See the Messages to Max topic for details.

There is no longer a general help patcher that appears when you choose Help from the Help menu and no object is selected. You could create a custom list of objects and descriptions and place it in the extras folder; it would then be listed in the Extras menu.

The **bpatcher** object no longer supports the replace message, which never worked reliably. You can use the new scripting features of the **thispatcher** object to accomplish the same task. See the new Max tutorials on Scripting for details.

The contextual menu for the **bpatcher** includes an item to open the original patcher. You can no longer command-option-click to get a small menu that lets you do the same thing.



The objects related to building patch librarians (**lib**, **libfrom** etc.) are no longer supported or included.

The **snd** object, which was unable to play sounds reliably at interrupt level, is no longer included.

The **glove** object, which worked with an interface called the GoldBrick that is no longer made, is no longer included.