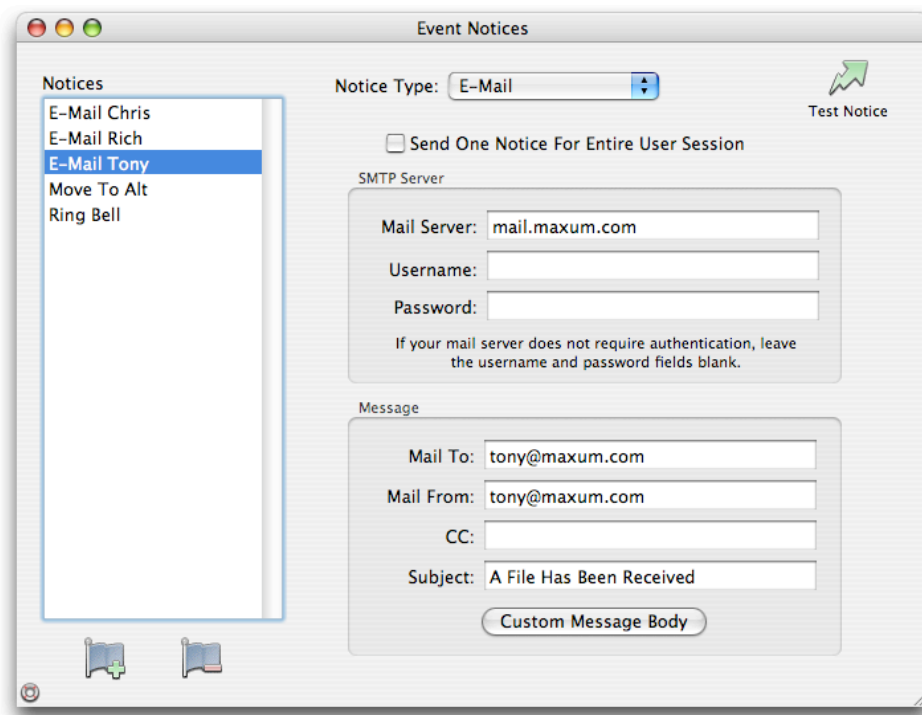


Event Notices

How to have Rumpus send you e-mail (or execute a script) when important events occur.

Using event notices, Rumpus makes it easy to have an e-mail sent to one or more people when files are uploaded to the server, downloaded from the server, or when other events occur. AppleScript or Unix shell scripts can also be executed to perform any other function, with a bit of scripting knowledge. You can create any number of event notices, each with one or several recipients, and then assign those notices to be triggered when individual users upload or download files, or when other events take place. To create or manage event notices, open the “Event Notices” window, shown below.



The Event Notices Window

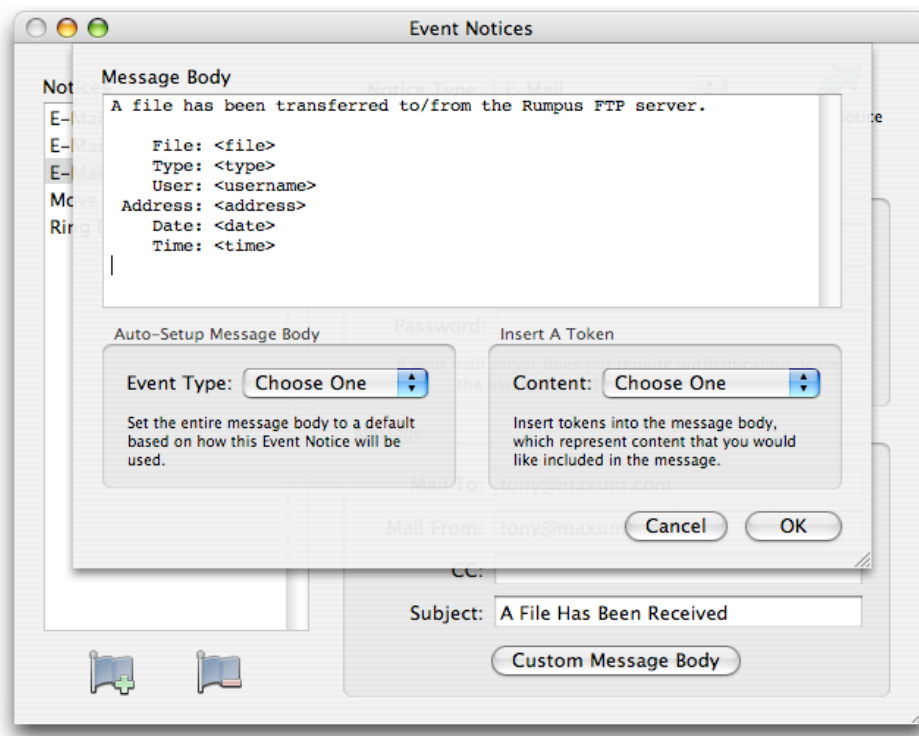
Defining Event Notices

After opening the Event Notices window, click the “Add Notice” button to create a new notice. A sheet will drop down, allowing you to supply a name for the new notice. Event notice names can be any easily recognizable text, and will be displayed in pop-up menus on the “Define Users” and other settings windows where you can select a notice for defined events. The “Add Notice” sheet also includes an option to start the new notice by duplicating the currently selected notice, if one is selected in the “Notices” list.

Creating An E-Mail Event Notice

If you would like the notice to send an e-mail message when triggered, start by setting the “Notice Type” to “E-Mail”. The fields that must be completed in order to have an e-mail sent are similar to those required when you set up any e-mail client. The server address is required, as is the “Mail To” (recipient), “Mail From” (sender) and “Subject”. For complete details on all of these settings, see the “Event Notices” help page in Rumpus.

By default, the e-mail message sent by Rumpus includes a very short message telling the mail recipient that a file has been sent, along with basic information about the file. If you like, you may customize this message by clicking the “Custom Message Body” button. When this button is pressed, the Custom Message sheet, shown below, will drop down.



Customizing The E-Mail Message Body

The large “Message Body” text area contains the e-mail message text, and can be edited just as you would any e-mail message. Information about the file transfer, user account, etc. is inserted into the message dynamically using tokens, the keywords surrounded by angle brackets, which are described below.

Automatic Message Bodies

The e-mail Event Notice message body defaults to a simple note describing a generic file transfer. However, since Event Notices can be used for a variety of purposes, other default messages can be generated for you automatically. In the “Auto-Setup Message Body” box, simply select the purpose of the Event Notice from the “Event Type” pop-up menu, and the existing message body will be replaced with a pre-formatted template.

Of course, the message body templates created using the “Auto-Setup” function can also be edited to suit your needs.

Dynamic Content Tokens

Within the message body, tokens are used to insert specific information about the file that was transferred. The easiest way to insert tokens is to place the cursor at the point in the message body where the file information should be included and then select the appropriate token from the “Content” pop-up menu.

The available tokens are:

<FILE> Insert the name and path of the file transferred.

<FILENAME> Insert only the name of file transferred.

<FILEURL UserAccount> Insert the file path, relative to the user account specified. This token is useful when including URLs to files in the e-mail message. Since the path to any particular file may be different depending on the Home Folder of the person accessing the file, the “UserAccount” parameter allows you to specify the account that will be used so that the URL generated will be correct for that user.

<FILESIZE> Insert the size of the file transferred.

<TYPE> Insert whether the transfer was performed via FTP or Web File Manager, and whether the file was uploaded or downloaded.

<USERNAME> Insert the username of the account being used to transfer the file.

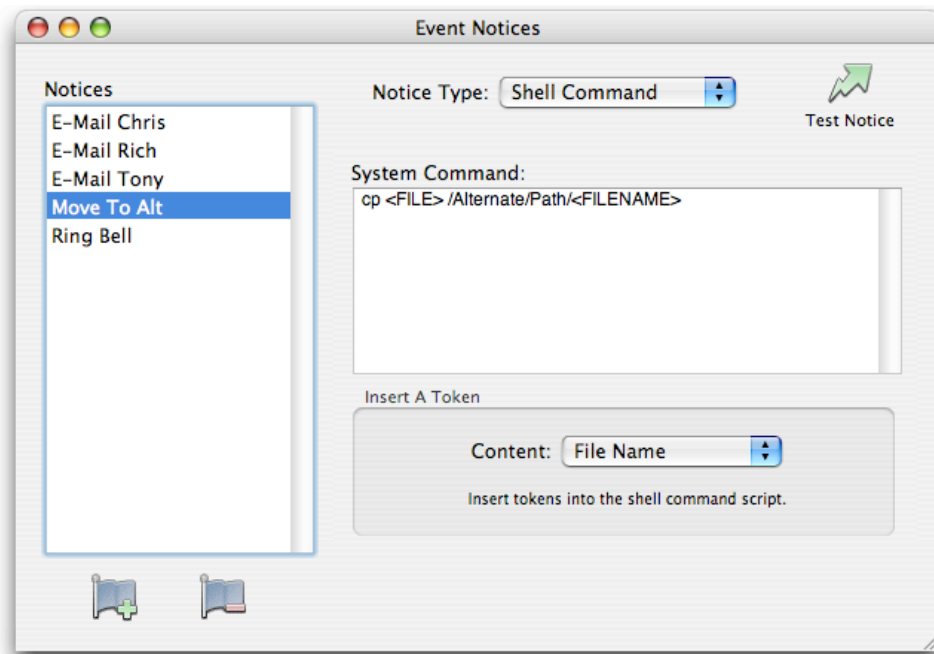
- <PASSWORD> Insert the user's password. Use of this token is not recommended for notices which will be used for authenticated user file transfers. It is provided exclusively for use with anonymous file uploads and downloads, where an e-mail address or other nonsecure password is supplied by the user for login.
- <EMAILADDR> Insert the e-mail address of the active user account, if it has been supplied on the "Define Users" window in Rumpus.
- <HOMEPATH UserAccount> Insert the Home Folder path of the specified user account. This token is primarily useful in scripts that perform some action within a user account folder. Note that if the "UserAccount" parameter is omitted, the token inserts the path to the active user account Home Folder.
- <ADDRESS> Insert the TCP/IP address of the client computer.
- <TIME> Insert the time the notice is sent (roughly the time the file transfer completes).
- <DATE> Insert the date on which the notice is sent.
- <FORMVAR VariableName> Insert the contents of a field included on the file upload form. This requires that you modify the WFM file upload form to include additional input fields. The "FormVar" token should then be edited so that "VariableName" matches the name of the input field on the form. This token is relevant only for Web File Manager file uploads.

Be sure to use the "Test Notice" button to send a test message. Rumpus will send the e-mail with sample values set for the filename, username, and so on and report any errors. If a known error occurs, Rumpus will suggest how to correct the problem, so be sure to read the error message carefully. If Rumpus reports no error, look for the e-mail in your usual mail program to confirm delivery.

When multiple files are uploaded but a single e-mail notice is sent, each file name and/or path will be reflected in the message. Any line in the message body that contains one or more of the tokens "File", "Filename", "FileURL" or "Filesize" will automatically be repeated for each file transferred.

Creating An AppleScript Or Shell Script Event Notice

To perform virtually any task when a file is transferred, use an AppleScript or Shell Script. Start by selecting “AppleScript” or “Shell Command” from the “Notice Type” pop-up menu. When you do, the fields used for defining the notice change to reflect the selected Notice Type. An example is shown below:



Shell Command and AppleScript notice types are each defined by a single text area, in which you can specify the script to be run. Like the e-mail custom message body, tokens can be used to have Rumpus insert common variables into your script. For example, the following shell script can be used to copy a file that has been uploaded to another location anywhere on your system:

```
cp <FILE> /AlternatePath/<FILENAME>
```

In this example, the “<FILE>” token will be replaced with the full path to the uploaded file, while the “<FILENAME>” token will be replaced by the filename. For example, if the file uploaded had a full path of “/Users/Shared/Test.pdf”, the resulting shell script would look like this:

```
cp /Users/Shared/Test.pdf /AlternatePath/Test.pdf
```

Simple AppleScripts can be created in a similar fashion. AppleScripts may be multiple lines, though long or complicated scripts should be created in the Script Editor application, saved as applets, and triggered by a simple script run by Rumpus.

A very simple AppleScript can be used to cause the system to beep when the event notice is triggered. This script would simply be:

```
beep
```

AppleScripts are actually run using the “osascript” system utility. Rumpus automatically parses each line of the script for variable tokens, and breaks up the script into the correct format for the osascript function.

It is important to note that both shell scripts and AppleScripts may not be able to be adequately tested using the “Test Notice” function, since test values used for the variable tokens won’t match actual files, user account information, etc. In addition, the script will be run in the context of the Rumpus control application, not the server daemon, which may also effect how the script is executed.

Let’s look at two more AppleScript based Event Notices. For example, if you want to display a message on the server itself, you could create an Event Notice to be triggered by file uploads that will tell the Finder to display a dialog box announcing the transfer. One problem with this is that the Finder will only display one at a time, and will hang notice processing while it waits for a user to accept the dialog. If your server doesn’t accept many files, this might not be a problem, so here's the script:

```
tell application "Finder"  
  activate  
  display dialog "File Upload: <FILE>"  
end tell
```

Another way to do something similar is to use the Stickies application to show uploads:

```
tell application "System Events"  
  tell application "Stickies" to activate  
  tell process "Stickies" to keystroke "n" using command down  
  tell process "Stickies" to keystroke "File Upload: <FILE>"  
end tell
```

These sample scripts will work fine with only minor changes, or can be extended to create more extensive server-side processes. As noted above, if your scripts become significantly more complicated, it is best to create stand-alone applets that are triggered by a one-line Event Notice script.