

# CHAPTER 35

## Event Scheduling

### IN THIS CHAPTER

ColdFusion Event Scheduling Concepts	171
ColdFusion Scheduling vs. Other Kinds of Scheduling	172
Creating, Modifying and Deleting Scheduled Tasks	173
Scheduling Application Examples	183

### ColdFusion Event Scheduling Concepts

The automated scheduling of activities is a part of everyday life. We schedule automatic monthly mortgage payments from our bank accounts, the recording of various television shows with our VCRs, notifications from our handheld devices and many other activities. By scheduling these events to occur in an automated fashion, we free up other resources to focus on issues that require our explicit interaction. It simply makes our lives easier knowing that the scheduled activity is going to get done without requiring our attention or time.

Of course, when we develop computerized systems to replace or enhance manual processes, we often want to schedule events to occur on a regular basis with the very same goals in mind. ColdFusion provides its own event scheduling mechanism that enables us to invoke one-time or recurring events. Through ColdFusion's scheduling mechanisms, you are able to invoke a specific ColdFusion template at a specific time. Because that template can be used to do just about anything that ColdFusion can do, this becomes a very powerful tool for the ColdFusion developer.

Here are some ideas for scheduled events:

- **Content management**—you can schedule the generation of static HTML pages from content that is stored in a database. This technique is commonly used in content management system to improve site performance
- **Automated e-mail applications**—you can schedule a process to check a POP3 server for e-mail and respond to it automatically with various prescribed messages or actions. This frees up the site administrator from a tedious task
- **Verity collection updates**—you can schedule the recurring update of the Verity collection that you've created for your site. This assures you that when visitors search your site, their results will reflect the site's current contents

ColdFusion provides two mechanisms for executing automated events or *scheduled tasks*:

- The Scheduled Tasks page in the ColdFusion Administrator
- The <CFSCHEDULE> tag

The functionality provided by the Add/Edit Scheduled Task page and the <CFSCHEDULE> tag is essentially the same. Both enable you to create, update, run, and delete tasks. However, you should be aware of a few distinctions:

- If you want to see a list of all the scheduled tasks on your server, you must use the Scheduled Tasks page in the ColdFusion Administrator.
- The Add/Edit Scheduled Task page is simply an HTML form. When you create or modify a task in the form, so it's a bit easier to work with than writing code.
- The <CFSCHEDULE> tag can be used without access to the ColdFusion Administrator.

It's important to recognize that when ColdFusion runs a scheduled task, it's run as if someone requested the URL (template) specified in the task. If you need a specific user to run that task (perhaps based on security concerns), you can specify the username and password. If output is to be generated by the task (perhaps for debugging purposes) you can specify that the output is to be published and then specify the file to which the output will be written.

→ These concepts will be discussed in more detail in the section "Creating, Modifying and Deleting Scheduled Events" in this chapter.

Also note that ColdFusion logs the execution of scheduled tasks in the scheduler.log file.

## ColdFusion Scheduling vs. Other Kinds of Scheduling

As you may already know, there are other ways to schedule automated tasks in computerized systems.

- Most advanced operating systems, such as Windows (XP, 2000, and XP), Unix, and Linux, provide their own mechanism for scheduling tasks.

In Windows 2000, NT, and XP you have the AT command (and there are some GUI utilities that work with AT as well).

- Many powerful database products, such as Microsoft SQL Server and Oracle, come with their own scheduling capabilities as well.

ColdFusion's scheduling capability is limited to requesting URLs through HTTP. With this in mind, there are situations in which one of the other approaches is more appropriate. For example, suppose you've built an index with the Verity Spider command-line utility (which is discussed in more detail in Chapter 29, "Full-Text Searching with Verity"). Because the Verity Spider runs from the command line prompt you could execute the command using <CFEXECUTE>, but it would probably be more appropriate to run this through the operating system. If you needed your database to dump some of its data in a delimited format to a data file every night, you could use a ColdFusion

template and a scheduled task to do this, but the database's native exporting tools and scheduling mechanism (assuming it had one) would probably be more appropriate.

Using ColdFusion's task scheduler has a couple of important benefits that are not necessarily provided by other scheduling approaches:

- The output from the specified URL can be saved to a file.
- The ColdFusion scheduler writes log entries each time a scheduled task is executed—and this log can be valuable in debugging.
- It can come in extremely handy if your application is hosted by an ISP that has limited your ability to run command-line programs or create scheduled tasks using their database server.

In this situation, consider using ColdFusion's event scheduling capabilities. Even if the ISP doesn't provide you access to the ColdFusion Administrator, you can use the `<CFSCHEDULE>` tag to run your task.

## Creating, Modifying and Deleting Scheduled Tasks

You can use either the ColdFusion Administrator or the `<CFSCHEDULE>` tag to create, modify, run and delete scheduled tasks. Your server's administrator may have decided to prevent developers from accessing the ColdFusion Administrator. If you don't have access, you can still get the job done programmatically using the `<CFSCHEDULE>` tag. If you do have access to the ColdFusion Administrator, you'll find this method preferable under most circumstances.

→ See Chapter 28, "ColdFusion Server Configuration," for a description of the ColdFusion Administrator usage.

### Administering Scheduled Tasks

The ColdFusion Administrator can be used to list, create, modify, delete, run and log the execution of scheduled events or tasks.

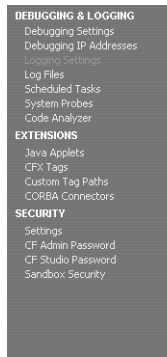
In earlier versions of ColdFusion, you could create a new scheduled task but ColdFusion would not recognize it until the scheduling service found the new task, and this would happen only in a pre-defined interval set up in the ColdFusion Administrator. This was referred to as "refresh interval." The refresh interval—and the interface you used in the ColdFusion Administrator to configure it—are gone in ColdFusion MX. In ColdFusion MX, new scheduled events are immediately recognized.

#### Scheduled Task Logging

You have the option of configuring the ColdFusion Administrator to log the execution of scheduled tasks. This setting is made at the bottom of the Log Settings page. Refer to Figure 35.1.

**Figure 35.1**

The “Enable logging for scheduled tasks” check box at the bottom of the Logging Settings page in the ColdFusion Administrator is used to enable scheduled task logging.



Maximum number of archives

Type in the maximum number of log archives Neo should create. After reaching this limit, files will be deleted in newest.

Use operating system logging facilities

When enabled, some ColdFusion log messages will be written using your operating system's logging facility (Windows NT; the syslog facility for UNIX.) Regardless of this setting, all ColdFusion log messages are also a standard ColdFusion log files.

Log slow pages taking longer than  seconds

To help diagnose potential problems or bottlenecks in your site, you can have ColdFusion log the names of a over a specified length of time to return. When enabled, any output will be written to the "server.log".

Log all Corba calls

Log all CORBA calls to help diagnose configuration. When enabled, any output is written to "server.log".

Enable logging for scheduled tasks

Log ColdFusion Executive task scheduling.

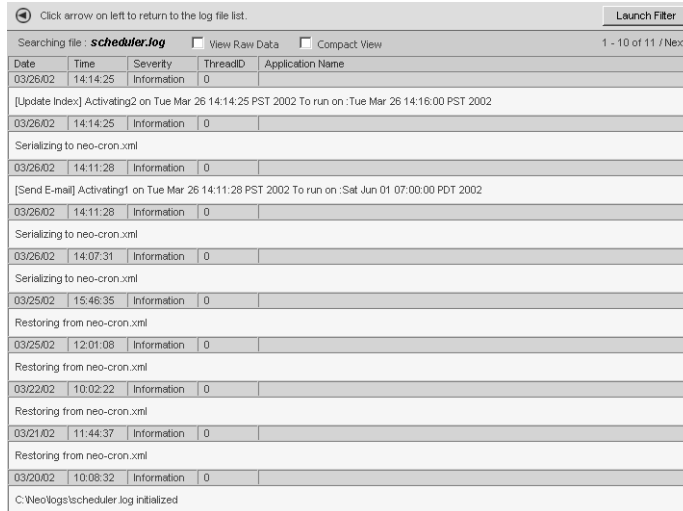
**NOTE**

Note that even if you don't enable logging of scheduled tasks, scheduled tasks that you manually execute will be logged.

Whenever a scheduled task is executed, an entry is written to the file scheduler.log in the Logs directory (below the directory in which ColdFusion is installed). (See Chapter 14, “Debugging and Troubleshooting,” for more information on ColdFusion’s logging capabilities.) You can review the entries in this log through the Log Files pages in the ColdFusion Administrator. See Figure 35.2. The log file itself is a simple quote and comma-delimited text file, as seen in Figure 35.3.

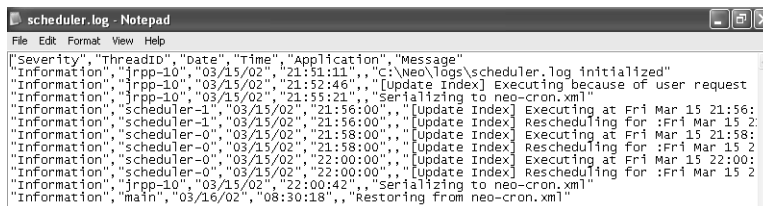
**Figure 35.2**

The scheduler.log file as viewed through the ColdFusion Administrator.



**Figure 35.3**

The scheduler.log file itself is a simple quote and comma-delimited text file.



The data presented in the log file is discussed in Table 35.1.

**Table 35.1** Interpreting the scheduler.log File

VALUE	DESCRIPTION
“Information”	Severity; in this case, simply informational
“scheduler-1”	Thread ID
“03/15/02”	Date of task execution
“21:56:00”	Time of task execution
<i>null value</i>	Application name. This is null in the scheduler log. It does not reflect the name of your ColdFusion application.
[Query Runner] Executing at Fri...	This message indicates the name of the scheduled task that is being referenced in this log entry.

As you browse through the scheduler.log file, you’ll note that ColdFusion makes one entry when the event is executed. If it’s a recurring event, you’ll see another to indicate the rescheduling of the event’s execution.

#### NOTE

Note that the log does not provide information regarding the success or failure of whatever underlying function is being provided by the task that you’re scheduling. Techniques for this kind of analysis are presented later in the section, “Scheduling Application Examples” in this chapter.

## Creating Scheduled Tasks with the ColdFusion Administrator

The ColdFusion Administrator’s Scheduled Tasks page provides a listing of all scheduled tasks, as seen in Figure 35.4. This listing is where you start when you want to create, modify, delete and run those tasks.

**Figure 35.4**

The Scheduled Tasks page in the ColdFusion Administrator is used to list, add, delete, modify and run scheduled tasks.

The screenshot shows the ColdFusion Administrator interface. On the left is a sidebar with a tree view containing categories like 'SERVER SETTINGS' (with sub-items: Settings, Caching, Client Variables, Memory Variables, Mappings, Mail Server, Charting, Java and JVM Settings, Manage Archives, Settings Summary) and 'DATA & SERVICES' (with sub-items: Data Sources, Verity Collections, Verity K2 Server, Web Services). Below that is 'DEBUGGING & LOGGING' (with sub-items: Debugging Settings, Debugging IP Addresses, Logging Settings, Log Files, Scheduled Tasks, System Error). The main content area is titled 'Scheduled Tasks' and contains the text: 'Scheduled tasks can create static web pages from dynamic data sources. You can also schedule tasks to update and to create reports.' Below this text is a 'Schedule New Task' button. At the bottom is a table with the following data:

Actions	Task Name	Duration	Interval
	Send E-mail	06/01/2002 - 09/01/2002	Weekly at 7:00:00 AM.
	Update Index	03/26/2002 - INDEFINITELY	Daily every 2 min(s) from 12:00:00 PM

Click the Schedule New Task button to create a new task. Doing so opens the Add/Edit Scheduled Task Form, as seen in Figure 35.5. Each field is explained in the following sections.

**Figure 35.5**  
The Add/Edit New Scheduled Task form.

### Name

This is the name that you want to give this task. It will appear in the list on the Scheduled Tasks page.

### Frequency

This part of the form enables you to specify the time frame in which the scheduled task operates. The time frame for execution can be:

- A one-time event
- A recurring event that occurs once a day, once a week or once a month
- A recurring event that occurs throughout the day

In any case you must at least enter a valid start time. When you are defining a task that must recur throughout the day, you must specify how often it recurs by entering hours, minutes or seconds (or a combination).

Select the appropriate radio button and enter the start time. If you are scheduling a task that occurs more than once a day, also enter the appropriate interval and if you like, enter an end time (after which the task will not be run again, until you reach the start time the following day).

### NOTE

Note that you can enter times in either 12- or 24-hour formats (i.e., 2:30:00 PM and 14:30:00 mean the same thing and either will work).

### TIP

It's important to recognize that each scheduled task that requests a URL from your server adds to your server's overall work load and can effect performance. Think carefully about the frequency with which each scheduled task must occur because you don't want to make your sever do a lot of unnecessary work.

## URL

This is the URL that you want ColdFusion to request, i.e. the task to be executed. You can run URLs using only HTTP; you can't use HTTPS, FTP or other protocols.

## Username and Password

If the URL you are requesting is protected by the web server's security mechanisms, you may need to authenticate with a valid username and password in order for the request to go through. Enter these values in these two fields.

### NOTE

Note that the only authentication that can be passed is basic clear text. If the server administrator has employed something more secure or proprietary, ColdFusion may not be able to authenticate when it executes this task. You can verify whether this is the case by saving the output of the URL request to a file. This technique is described in the section "Publish and File" in this chapter.

## Timeout (sec)

If you are requesting a URL that is complex or may take a while to complete, you may need to tell ColdFusion to give this request a longer timeout.

### NOTE

Note that ColdFusion's default request timeout duration is set on the Settings page in the ColdFusion Administrator.

→ See Chapter 28, "ColdFusion Server Configuration," for a description of this setting.

In this situation, you can enter a longer time period, in seconds.

## Proxy Server (and port)

If your ColdFusion server goes through a proxy server to reach to the URL you're requesting, enter the proxy server and port information in these to fields.

## Publish and File

If you want the output from the task to be saved to a file, check this check box and enter the path-name of the file in the File field (e.g., c:\temp\output.htm).

### TIP

This feature is useful for debugging the execution of your task. If your task is something that doesn't really produce output for publishing (like HTML), you may still want to use this feature to write a date and time stamp or some other debugging information to a file to give you an indication of what happened with your process.

## Resolve URL

If you are requesting a URL from an external site but you need to be able to use its internal references (perhaps you're requesting a URL from an external website that you plan on publishing on your server), you should check this box to resolve the URLs that are used within the page you're requesting.

**TIP**

When you're done creating a task, be sure to use the Submit button at the bottom of the form to save the new task.

## Running, Modifying, and Deleting Scheduled Tasks

The Scheduled Tasks page in the ColdFusion Administrator lists all of the scheduled tasks in your ColdFusion server. There are three icons at the left of each entry (see Figure 35.4). The first icon is for running the task, the second is for modifying it and the third is for deleting it.

### Running Scheduled Tasks

To run the task now, click the leftmost icon, labeled “Go”. This ad hoc method of executing a scheduled task is most helpful when you're getting your task set up and debugging the process.

**TIP**

If you configured the task to publish its output to a file, you can open that file after running the task to see what happened. If the URL you entered was incorrect or there was some other problem requesting it, you should have enough valuable information in the saved output to debug the situation.

### Modifying Scheduled Tasks

The second icon at the left of the task list (in the Scheduled Tasks page in the ColdFusion Administrator) is used to modify the configuration of that task. You can also just click on the task's name in the list. Clicking either the icon or the task name loads the Add/Edit Scheduled Task page, as shown in Figure 35.5.

The fields all work just as they did when you created the task. Make whatever changes are required and save the task using the Submit button at the bottom of the form.

**NOTE**

Note that the act of saving a scheduled task like this creates its own entry in the scheduler.log file. This entry indicates that the task is being activated.

### Deleting Scheduled Tasks

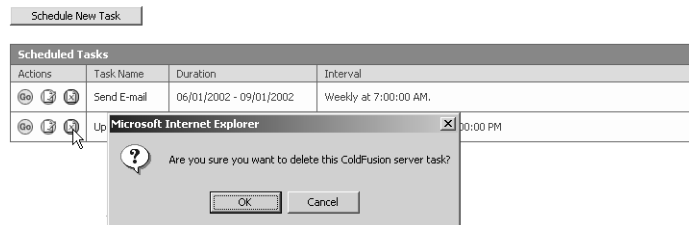
When you determine that you no longer need to keep a scheduled task, you can delete it by clicking the rightmost icon (of the three) in the Scheduled Tasks list. Having no-longer-scheduled tasks in the list doesn't hurt anything, but if there are a lot that are no longer running, it may make sense to delete some of them to make the list easier to read.

When you click the delete icon, you'll be prompted to confirm that you do indeed want to delete the task (see Figure 35.6). Assuming you do, click the Ok button, the task is removed from the list and the list is refreshed.



**Figure 35.6**  
Deleting a  
scheduled task.

Scheduled tasks can create static web pages from dynamic data sources. You can also schedule tasks to update Verity searches and to create reports.



### Creating, Modifying, and Deleting Tasks Using <CFSCHEDULE>

As explained in the section “ColdFusion Event Scheduling Concepts,” you can use either the ColdFusion Administrator or the <CFSCHEDULE> tag to manage scheduled tasks. You’ve already learned how to employ this functionality using the ColdFusion Administrator. This section demonstrates the use of <CFSCHEDULE> to accomplish the same things.

The <CFSCHEDULE> tag is commonly used when your server administrator has configured the ColdFusion Administrator so that you don’t have access to the Scheduled Tasks list. When you can’t use the ColdFusion Administrator’s form-based interface, the <CFSCHEDULE> tag comes in very handy.

**NOTE**

Note that you actually could build a form-based interface to the <CFSCHEDULE> tag. You could start by copying the HTML code from the ColdFusion Administrator and saving it somewhere that allows users access. You’d then have to build a template to validate the user’s input. Assuming the input was good, you’d feed it to the <CFSCHEDULE> tag. Using this approach, the ColdFusion Administrator would remain protected and users would still have an easy-to-use, form-based interface for creating scheduled tasks.

### Creating a Scheduled Task with <CFSCHEDULE>

Table 35.2 lists and describes the <CFSCHEDULE> tag attributes. Note that items marked with an asterisk (\*) are required only under certain circumstances. The specific cases are noted in the description of the attribute.

**Table 35.2** <CFSCHEDULE> Tag Attributes

ATTRIBUTE	REQUIRED	DESCRIPTION
TASK	Yes	The name of the task to perform the action on.
ACTION	Yes	DELETE, UPDATE, or RUN. Specifies which action is required of the task scheduler. Note that UPDATE produces a new task if the named task doesn’t already exist. If the named task does exist, using UPDATE will overwrite that task. RUN executes the task now and DELETE deletes it now.

**Table 35.2** (CONTINUED)

ATTRIBUTE	REQUIRED	DESCRIPTION
OPERATION	Yes*	The type of operation to be performed but, can be set only to "HTTPRequest". Required only if the ACTION attribute is UPDATE.
STARTDATE	Yes*	The date on which you want the scheduled event to begin. This attribute is required when the ACTION attribute is UPDATE.
STARTTIME	Yes*	The time at which you want the scheduled event to begin executing. This attribute is required when the ACTION attribute is UPDATE. Provide a complete time value in 12- or 24-hour format.
URL	Yes*	The URL to be requested when the task is executed. Required when the ACTION attribute is set to UPDATE.
INTERVAL	Yes*	The interval at which the scheduled task is to be executed. Valid values are ONCE, DAILY, WEEKLY, MONTHLY or a time interval in seconds (an integer value). This is required only when the ACTION attribute is set to UPDATE. Use seconds to specify intervals that recur throughout the day (e.g., every hour = 3600).
PUBLISH	No	Indicates whether the results of the URL request are to be saved to a file. Valid values are YES and NO.
PATH	Yes*	The path, ending with "\", where the file specified in the FILE attribute is to be created. This is required only if PUBLISH is set to YES.
FILE	Yes*	The filename for the results of the scheduled task. This is required only when the PUBLISH attribute is set to YES.
ENDDATE	No	The date on which the scheduled task should stop being executed.
ENDTIME	No	The time at which the scheduled task should stop being executed. You can use either 12- or 24-hour time formats.
REQUESTTIMEOUT	No	The request timeout period (in seconds) that should be used for this request.
RESOLVEURL	No	Indicates whether you want the internal URLs (in the content of the requested URL) to be resolved so that they work when this page is loaded from a server other than the original server.
USERNAME	No	The user name to use to authenticate when requesting a protected URL.
PASSWORD	No	The password to use to authenticate when requesting a protected URL.

**Table 35.2** (CONTINUED)

ATTRIBUTE	REQUIRED	DESCRIPTION
PROXYSERVER	No	The hostname or IP address of a proxy server. This is required only if your ColdFusion server needs to use a proxy server to reach the requested URL.
PROXYPORT	No	The port used by your proxy server. It is used only if your proxy server works on a port other than 80.
PORT	No	The port to be used by your server when executing this task.

When you're using <CFSCCHEDULE> to create scheduled events, you set the ACTION attribute to UPDATE. Suppose you wanted to create a scheduled task that involves updating the Verity index of your site. First you'd write the ColdFusion template that would update the Verity index. In this simple example, let's assume that this template is called collection\_update.cfm and is located in a directory off the web server's root named "/tasks". The task will be named "Update collection". You want to keep the Verity collection up to date so you want this task to run nightly. Because you don't want it to run when there's a lot of traffic on your server, you want it to run every night at 11:30 PM. See the <CFSCCHEDULE> tag that you could use for in Listing 35.1.

**Listing 35.1** Using <CFSCCHEDULE> to Create a New Scheduled Task

```
<CFSCCHEDULE
  ACTION="UPDATE"
  TASK="Update collection"
  OPERATION="HTTPRequest"
  URL="/tasks/collection_update.cfm"
  STARTDATE="04/01/2002"
  STARTTIME="11:30:00 PM"
  INTERVAL="Daily"
>
```

After executing this code and opening the ColdFusion Administrator, you can see the resulting scheduled task displayed in the form, as shown in Figure 35.7.

As you can see in Listing 35.1, you do not have to write a lot of code to create a recurring task. Let's review what each attribute does.

- ACTION="UPDATE"—Tells ColdFusion to update an existing task with the name specified in the TASK attribute. If that task does not exist, ColdFusion will create a new one.
- TASK="Update collection"—Tells ColdFusion the name of the task to update (or create, as in our case).
- OPERATION="HTTPRequest"—Tells ColdFusion that this request is to be made through HTTP (this is the only way that requests currently can be made).
- URL="/tasks/collection\_update.cfm"—Indicates the URL to be requested.
- STARTDATE="04/01/2002"—Indicates the date on which the task is to be first executed.

**Figure 35.7**

This scheduled task was created by the code in Listing 35.1.

- STARTTIME="11:30:00 PM"—Indicates the time at which the task is to begin running.
- INTERVAL="Daily"—Indicates the time interval at which this event is to recur.

A more detailed example will be provided below.

### Modifying, Deleting, and Running Scheduled Tasks with the <CFSCHEDULE> Tag

Modifying scheduled tasks through use of <CFSCHEDULE> works virtually the same way as creating a scheduled task with <CFSCHEDULE>. The previous section provided a simple example of its usage for creating a scheduled task.

Suppose after updating the Verity collection nightly for a month, it became apparent that you don't need to update the collection more than once a week. The following code would make this modification (see Listing 35.2).

**Listing 35.2** This Code Modifies an Existing Task

```
<CFSCHEDULE
  ACTION="UPDATE"
  TASK="Update collection"
  OPERATION="HTTPRequest"
  URL="/tasks/collection_update.cfm"
  STARTDATE="04/01/2002"
  STARTTIME="11:30:00 PM"
  INTERVAL="Weekly"
>
```

Deleting a scheduled task with <CFSCHEDULE> is simply a question of setting the ACTION to Delete and naming the task:

```
<CFSCHEDULE ACTION="Delete" NAME="Update collection">
```

To execute a scheduled task immediately, use this code:

```
<CFSCHEDULE ACTION="Run" NAME="Update collection">
```

## Scheduling Application Examples

You're now ready to put your new understanding of ColdFusion's scheduling capabilities to use. First, you'll create a scheduled task for POP3 application (List Unsubscriber) presented in Chapter 26, "Interacting with E-Mail." Then you'll build two simple scheduling applications. The first application sends e-mail out on a regular basis. The second updates the Verity index that gets built in Chapter 34, "Full-Text Searching."

### Creating a Scheduled Task for a POP3 Application

First, please quickly review the POP3 application developed in Chapter 26. This application checks a POP3 e-mail account and looks for incoming messages to the mailings@orangewhipstudios.com account in which the subject includes the word "remove". When a message like this is found, the template identifies the sender in the Contacts table and sets the value of MailingList field to 0, flagging it as an account that should not have promotional e-mail sent to it.

When you run this template, it will operate on the messages it finds in the mailings@orangewhipstudios.com account. But running this template only once is of limited usefulness. Moreover, you don't want to rely on a person to remember to execute this task from time to time. The ColdFusion task scheduler is ideally suited for this application.

Note that people may unsubscribe throughout the day. And yet, OWS staff may continue to send spam, ahem, make that "promotional e-mail" throughout the day. So we want to find a schedule that minimizes the opportunity for someone to unsubscribe and then wind up with three new e-mails from OWS. At the same time, we don't want to bog down ColdFusion by running this script every 10 seconds. Once an hour seems like a good compromise.

Listing 35.3 creates a scheduled task to run the ListUnsubscriber.cfm template from Chapter 26.

#### Listing 35.3 ListUnsubscribe\_Task.cfm

```
<!---
Name:          ListUnsubscribe_Task.cfm
Author:        Leon Chalnick
Description:    Creates a scheduled task to
                Run ListUnsubscriber.cfm
                once per hour starting at
                2:00:00 AM on 3/18/2002.
Created:       3/18/2002
-->
<CFSCHEDULE
    TASK="Mailing list unsubscriber"
    ACTION="UPDATE"
    OPERATION="HTTPRequest"
    URL="http://localhost/ows/26/ListUnsubscriber.cfm"
    PORT="8500"
    INTERVAL="3600"
    STARTDATE="03/18/2002"
    STARTTIME="02:00:00 AM"
>
```

Note that the value of the URL attribute employs port 8500, indicating use of ColdFusion's local web server (as opposed to another web server like MS IIS). If you're using a different web server, for example, one that uses port 80, you will not need to include the 8500 in the PORT attribute.

This task will begin running at 2:00 AM each night and will run on the hour. Please refer to Figure 35.8.

**Figure 35.8**  
The Scheduled task created by Listing 35.3.

The screenshot shows the 'Add/Edit Scheduled Task' dialog box. The 'Task Name' field contains 'Mailing list unsubscriber'. The 'Duration' section shows 'Start Date' as 3/18/2002 and 'End Date (optional)' as empty. Under 'Frequency', the 'Daily every' option is selected. The 'Hours' field is set to 1, 'Minutes' to 0, and 'Seconds' to 0. The 'Start Time' is 2:00:00 and 'End Time' is empty. The 'URL' field contains 'http://localhost:8500/ows/26/ListUnsubscriber.cfm'. There are empty fields for 'User Name', 'Password', 'Timeout (sec)', and 'Proxy Server'. The 'Publish' checkbox is unchecked, and the 'File' field is empty. The 'Resolve URL' checkbox is also unchecked. 'Submit' and 'Cancel' buttons are at the bottom right.

## Building an Automatic Promotional E-mail Application

Let's assume that OWS's marketing department wants to capitalize on the large media blitz underway to support the new releases. It wants to send out promotional e-mails to local memorabilia dealers once a week that list the current merchandise available at the online store. To make matters tougher, the marketing department wants the e-mail to be waiting in the dealers' in-boxes when they first check their e-mail each Monday morning.

Therefore, you must build a ColdFusion scheduling application that has two parts:

- A page that queries the database for the current merchandise available, displays the results in an HTML table and sends the e-mail page to the dealer group e-mail alias.
- A scheduled task page that sends the promotional e-mail once each week on Monday morning.

### Building the E-mail Application

The e-mail application consists of two parts: the query that collects the data and the <CFMAIL> tag that sends it. See Listing 35.4 for this part of the application.

**Listing 35.4** Promo\_Email.cfm—The E-mail Application

```

<!--
Name:          Promo_Email.cfm
Author:       David Golden, Leon Chalnick
Description:  Queries database and generates e-mail
Created:     4/15/01
Modified:    3/17/2002
-->

<!-- Get all merchandise info from database -->
<CFQUERY NAME="GetMerchInfo" DATASOURCE="OWS">
    SELECT MerchName, MerchDescription, MerchPrice
    FROM Merchandise
    ORDER BY FilmID
</CFQUERY>

<!-- Produce the e-mail -->
<CFMAIL TO="dealers@ows.com" FROM="store@ows.com" SUBJECT="Merchandise Currently
Available" SERVER="mail.ows.com" TYPE="HTML" GROUPCASESENSITIVE="yes">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
    <TITLE>Available Merchandise From OWS Online Store
    </TITLE>
</HEAD>
<BODY>

Greetings from Orange Whip Studios!

Here is a list of merchandise currently
available from the online store at Orange Whip
Studios. To order any of these items, go to
<A HREF="http://www.ows.com/store/">http://www.ows.com/store/
</A>.

<!-- Display merchandise from query -->
<TABLE BORDER="1">
<TR>
    <TH>Item name</TH>
    <TH>Description</TH>
    <TH>Price</TH>
</TR>
<CFOUTPUT QUERY="GetMerchInfo">
    <TR>
        <TD>#MerchName#</TD>
        <TD>#MerchDescription#</TD>
        <TD>#DollarFormat(MerchPrice)#</TD>
    </TR>
</CFOUTPUT>
</TABLE>
</BODY>
</HTML>
</CFMAIL>

```

The `<CFQUERY>` tag selects all the merchandise in the database. The `<CFMAIL>` tag sends the mail to an e-mail list of dealers. Notice the use of the `TYPE` attribute in the `<CFMAIL>` tag. By setting it to “HTML”, you are able to send nicely formatted e-mail.

**NOTE**

Note that when sending e-mail formatted with HTML, the recipient must be using an e-mail client that supports HTML e-mail (such as Microsoft Outlook). Another way you could send this listing would be as an attached HTML file. To do this, you would separate the production of the HTML merchandise listing from the sending of the e-mail. You could create a separate scheduled task to run this new HTML page and save its output to a file. Another scheduled task would then be executed to send the mail and you would use the `<CFMAILPARAM>` tag to attach the new HTML listing output created by the first new scheduled task.

The `<CFOUTPUT>` within the `<CFMAIL>` is used to output the query results in the body of the message.

### Scheduling the Task

The second part of this application requires scheduling the task. In this sample application, you'll use your knowledge of the ColdFusion Administrator Scheduled Tasks page to accomplish this.

In the ColdFusion Administrator, click on Scheduled Tasks and view the Scheduled Tasks list, as seen in Figure 35.4 (note that the list may contain different scheduled tasks on your ColdFusion sever). Click the Schedule New Task Button and open the Add/Edit New Scheduled Task form. Complete the form as seen in Figure 35.9.

### Scheduling Updates to a Verity Collection

OWS wants web site visitors to be able to do full-text searches of their website's content. A Verity collection was created to these ends (see Chapter 29, “Full-Text Searching with Verity,” for more information about Verity collections). In a nutshell, a Verity collection is a searchable index of one or more documents that you build in ColdFusion (either through the ColdFusion Administrator or through the `<CFCOLLECTION>` tag). You can then search these documents for words or phrases using the `<CFSEARCH>` tag.

As the content of OWS's site changes from time to time, it is important to update the Verity collection to reflect these changes. It has been determined that this should take place nightly at 11:30 PM. As with most scheduled tasks, there are two parts to the solution. You must create the code to be run by the scheduled task and you must create the task itself.

### Updating the Verity Collection

You can programmatically update a Verity collection through ColdFusion with the `<CFINDEX>` tag. The collection of the OWS site is created in Chapter 34, “Full-Text Searching”. Here, we're simply going to produce a simple template to update it. See Listing 35.5 for the code that updates the Verity collection.



**Listing 35.5** OWS\_Verity\_Update.cfm—Updating the OWS Verity collection

```

<!---
Name:          OWS_Verity_Update.cfm
Author:       Leon Chalnick
Description:   Updates the OWS Verity collection
Created:      3/18/2002
-->

<!--- Update the OWS collection --->
<CFINDEX COLLECTION="ows" ACTION="update" TYPE="path"
  KEY="C:\cfusionmx\wwwroot\ows\34\HR\Docs">
<!--- Create debugging output to display last update of OWS
  collection --->
<CFOUTPUT>
OWS collection last updated: #DateFormat(now())#
#TimeFormat(now(), "h:mm:ss tt")#
</CFOUTPUT>

```

Note that a little text is being output at the bottom of Listing 35.5. In the next section, you'll create a <CFSCHEDULE> tag to create the task. You'll configure this scheduled task so that the output is saved to a file. This debugging comment will be written to the specified file each time the task is executed.

**Creating the Scheduled Task with <CFSCHEDULE>**

Rather than create this task through the ColdFusion Administrator as you did in the previous example, this time you'll create it through code. See Listing 35.6. The task must be scheduled to run each night at 11:30 PM. It must also save its output to a file named ows\_verity\_update\_output.htm in a folder named c:\temp\.

**Listing 35.6** Sched\_Verity\_Update.cfm—Creating the scheduled task to invoke OWS\_Verity\_Update.cfm

```

<!---
Name:          Sched_Verity_Update.cfm
Author:       Leon Chalnick
Description:   Creates a scheduled task to
              update the OWS Verity collection
              nightly and write output to file.
Created:      3/18/2002
-->
<CFSCHEDULE
  TASK="OWS Verity Update"
  ACTION="UPDATE"
  OPERATION="HTTPRequest"
  URL="http://localhost/ows/35/ows_verity_update.cfm"
  PORT="8500"
  INTERVAL="Daily"
  STARTDATE="03/18/2002"
  STARTTIME="11:30:00 PM"
  PUBLISH="Yes"
  PATH="c:\temp\"
  FILE="ows_verity_update_output.htm"
>

```

When you run this template through your browser, it instructs ColdFusion to create this scheduled task.

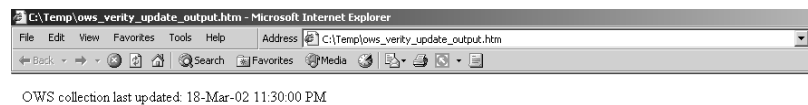
**NOTE**

Note that when this task is run, it won't work unless you have already created the OWS Verity collection and have a directory named `c:\temp\`.

When this scheduled task does actually run, it will create the output file, `ows_verity_update_output.htm` in the `c:\temp\` folder. The contents of this file, after the “OWS Verity Update” scheduled task ran on March 18, 2002, are displayed in Figure 35.9.

**Figure 35.9**

The output from running the OWS Verity Update scheduled task.



OWS collection last updated: 18-Mar-02 11:30:00 PM