

COLD FUSION Developer's Journal

ColdFusionJournal.com

June, 1999 Volume: 1 Issue: 3

Guest Editorial

Enter the Scalability Zone

Steve Drucker page 5

Tips & Techniques

The Black Box Technique

Dan Chick page 24

CF Tags

An Extension to ColdFusion

Paul Underwood & Simon Clarke page 36

Tips & Techniques

Creating Subforms with CFGRID

Mark Cyzyk page 38

CF & LDAP

Connecting CF to LDAP

David Anderson page 46

CFDJ News

Allaire Acquires Bright Tiger Technologies

CFDJ Marketplace page 47

RETAILERS PLEASE DISPLAY UNTIL JULY 31, 1999

SYS.CON
PUBLICATIONS



The Problem of Complexity

Complex projects call for methodical measures

Feature: From Zero to Finish in Three Months

Launching documents around the globe calls for high-tech control measures



6

Leon Eno

<BF on CF>: Ben Forta on ColdFusion

Take your database out of retirement



12

Ben Forta



IMHO: Building Enterprise Portals

Finally, a way to think about how complex technology and business tie together cohesively



50

Jeremy Allaire

Feature: Streamlining Data Entry Applications

Standardize your maintenance techniques to make your life easier



16

Leon Chalnick

Special Feature:

The Problem of Complexity PART 1
Complex projects call for methodical measures



28

Hal Helms, Steve Nelson & Gabe Roffman

CF Basics: More Punch with <CFQuery>

The challenge: To allow site visitors to text-search a database and display results in a table



44

Jerry Bradenbaugh





FROM ZERO FINISH THREE

TO IN MONTHS

by
Leon Eno

Launching documents around the globe calls for high-tech control measures

Anyone involved with meeting tight deadlines quickly realizes the power of ColdFusion. When my marketing organization asked me (Leon) to create a system to assist in revision control for new-product launch materials around the world, I wasn't given much time to do it. Getting large numbers of documents from diverse geographic locations to everyone involved in a product launch concerns any corporation whose business is global in nature. One of the main problems in developing such a system is the ramp-up time required to obtain the necessary tools and skills, particularly if this involves learning a new language.

My experience as a Web programmer prompted my management to request that I obtain all documents from the parties involved in the launch process, open and save the documents in Office 95 format and FTP the files to our server. I then had to code

an HTML page to point to the individual files. As we all know, this process is an exercise in futility. After notifying all locations of the "hidden" URL, I'd begin accepting documents and post them to the URL. It didn't take much time to realize I was connected at the hip to this "little" maintenance job (which, incidentally, I wanted no part of).

I decided there had to be a better way. Although I have over 20 years of coding experience, I'd never coded in a GUI environment. Nevertheless, I was quite aware of what could be accomplished. Since laziness is the mother of invention (maybe a few of you can relate to this), I sat down and analyzed the situation, then drew up a spec on how this revision control system could be automated. Clearly I had to remove myself from the constant updating and refreshing of information.

With spec in hand I went to a colleague and asked if he could create a prototype in CGI or PERL script within a day or so. I was virtually laughed out of the office. At this time Rob, a young contractor working in my organization, suggested a product called ColdFusion. We discussed the spec – a secure area that only authorized individuals could access; and the need to upload data from an individual’s desktop, automatically e-mail associates with a detailed comment and create an area with revision history. Rob researched ColdFusion, then came back to me the next day and said it was all possible. We picked up a copy of ColdFusion Server that evening and Rob had a prototype together in a matter of a few days. I was excited; we’d found a tool to implement my spec and I was saved from this never-ending, tied-to-the-hip task. Life was good. I could now turn this project around in a very short period of time and get on with my other duties.

After a week we were making significant progress and were able to demo the prototype to management on Friday. Management liked what they saw and agreed to fund the development as long as it could be accomplished in one month. A month? Ha! With Rob here and my ability to nudge a little extra effort, there was no problem.

Come Monday morning we were ready to really attack this project. Rob arrived, got his coffee and proceeded to give notice. Aargh! Now what? I certainly wasn’t going to have time to implement this, along with all my other duties. After a short search, I found George Pandapas, a ColdFusion expert, and the rest, as they say, is history. In the course of the next couple of weeks, the system was functioning in a basic operational manner and by the end of the third month we had something appropriately named for its acronym: W.I.N., or Workstation Information Network.

W.I.N. – What Is It?

The Workstation Information Network is an automated revision control system designed to organize, store, authenticate and distribute the most recent documents and files pertaining to workstation products. The system provides a simple, effective, automated, worldwide, Web-based system that’s available to appropriate workstation personnel. Before W.I.N., revision control was manual and chaotic. With the new system, the workstation team has access to the most recent documentation immediately after the person in charge of content creation has uploaded it. Various documents are neatly organized and centrally located. W.I.N. is dynamic and reflects changes immediately.

W.I.N. was designed to assist in maintaining revision control for projects such as new-product announcements and as a nondisclosure information repository. It allows authorized personnel to access the databases specific to the projects they’re assigned to, and to view, update and upload new documents into the database. The system generates automatic e-mail to all participants of a specific project with information pertaining to the nature of the uploaded file and with a comment supplied by the individual who uploaded it. It’s Web-based and easy to use. It enables a review of draft and final versions of documents, and is password-protected to safeguard sensitive information.

What Are Its Benefits?

Content providers for the workstation’s business are ensured that all participants in a project are aware of the most up-to-date version of their document. It replaces the need to e-mail large files to a host of individuals who are in need of the information. It easily enables distribution of drafts for review to the appropriate areas and all Product Marketing and Development (PM&D) members, resulting in worldwide immediate access to updated documents.

Who Is the Audience?

The audience consists of content creators for the workstation business, primarily residing in the PM&D. Other users are geographical-area marketing personnel and sales team members. All users are notified of the availability of draft and final documents. It facilitates the review of documents before they are final, and allows for last-minute updates. Participants on a project can download, localize and re-post information locally on their server. Other PM&D members can benefit from W.I.N. by receiving final documents faster than before.

What’s the Learning Curve?

A user is virtually an expert after one login. W.I.N. has been designed to be user-friendly, and is very functional in its simplicity. The main page used to enter the site acts as an introduction. From this page you can access the W.I.N. online instructions or guidelines, and move to the login screen to enter the site. All navigation options are presented via buttons at the top of the screen, under the W.I.N. title banner.

The system requires each new user to create a unique password upon first entry to the site. As in all password-protected systems, it is important to keep in mind that the users be instructed as follows: Keep in mind that an administrator will never ask you for your password. Don’t share it with anyone.

If an incorrect password is entered, the user is notified and directed back to the login screen via a meta refresh statement in HTML code.

Upon successful login, the user is requested to select the project they’d like to work on, if they’re registered for more than one (see Figure 1). After your project is determined you’ll be transferred to the main, “spreadsheet” page.

The main W.I.N. page lists the documents for a specific project. The document name is linked to the actual document file, and the user can view/download it by clicking on the link. The author’s name is also linked. By clicking on this link you can send a comment to the author via e-mail. The status column contains pull-down menus listing whether the most recent document is a draft or final version. The document’s author can change the version via a pull-down menu. The status pull-down menu appears for all documents for which the user is the author.

This routine has probably the most complex SQL statement in the system. The first task is to retrieve the latest version of each document in the currently selected project using a subquery involving the “revision” field, as noted below. Next, the document and author tables are joined to obtain the appropriate author information, except in the case of the “ARCHIVE” project, which we use to logically delete or deactivate documents that may be reactivated later. In the latter case we had to dynamically alter the WHERE clause to prevent the “disappearance” of archived documents because no authors are registered in this project. Next, we provide a sort by column feature by dynamically altering the ORDER BY clause, with an additional toggling effect achieved by checking the Client variable list for the column’s sort parameter and reversing its sign if found (see Listing 1).

The spreadsheet page displays important information for each document. The first 30 characters of the comments associated with each document are displayed. Each comment is hyperlinked so that if it exceeds 30 characters it can be clicked to reveal the entire comment in a JavaScript-generated window. The revision history button will take you to the Revision History Page. The Date column displays the creation date in European format for its associated document. This is not necessarily the date the document was uploaded, but rather the date the document was originally created. The Revision column displays the document’s latest version number, and the Size column displays the size of the document. Other visual features include highlighted rows for documents less than two days old and column headings that repeat every 10 rows so users always know what they’re viewing as they scroll vertically.

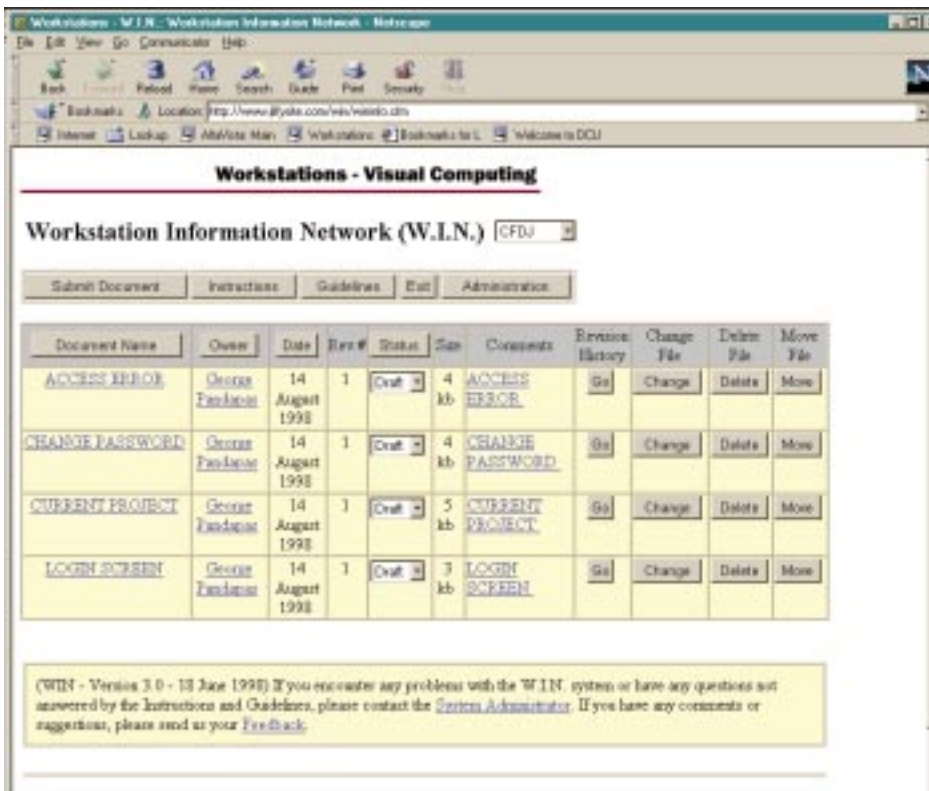


Figure 1: The W.I.N. spreadsheet

The document upload form (see Figure 2) is used to upload documents to a predetermined location on the server's file system. The "Browse" button is included so users can conveniently locate the file on their local system. Clicking this button displays users' folders and files in a manner similar to the Windows Explorer interface, and permits them to click through the hierarchy of folders until they have located their document file. Once located, they upload it by simply selecting it and clicking the "Open" button on the navigation window. The browse window will then close and the file's path will be displayed in the "File To Upload" text field. The "Document Name" field contains a descriptive title associated with the file, which will appear on the W.I.N. spreadsheet page. When uploading a revision of an existing document, the user must use the filename used previously. To assist in this task, there's a list of document and associated filenames on the right. This is a variable-length list with a maximum length of 10 rows (to stay within the dimensions of the HTML table). Listing 2 provides the code to generate this list.

The document upload acknowledgment page offers a summary of the document just uploaded. If the document file was already on the server, it's been overwritten by the new version submitted. The summary page template also sends e-mail to everyone on the distribution list associated with the project. Additionally, this template will handle duplicate filename conditions by continual-

ly prompting users to rename their file until it becomes unique. This condition occurs when the user attempts to upload a file with the same name as a preexisting file owned by another user. When you're not the owner of the existing file, W.I.N. won't permit you to overwrite it as a document revision. From this page the user has the option to navigate back to the W.I.N. page, which will now display all the information for the new document just uploaded.

Administration of W.I.N.

The W.I.N. administrator is responsible for ensuring that all users are added to the appropriate projects, and that their registration information is properly maintained. In addition to user administration, the administrator may create, change and delete projects. The administrator also has sole access to the following buttons on the W.I.N. spreadsheet page: Administration for entering the administration area, Change for modifying document information, Delete for deleting documents and Move for transferring documents to other projects, including the reserved "ARCHIVE" project for temporarily "deactivating" certain documents.

The Administrator can add, change or delete a single user, multiple users and projects. He or she can copy users from one project to another, list all users either in one project or in the entire system, and subsequently print these listings. The Administrator can turn everyone's e-mail off or on, and can edit the author table record by record.

To add a user, select a project and click on the Single User Add button (see Figure 3). The selected project will be highlighted in a list of all W.I.N. projects. The Administrator may change this selection and/or select additional projects.

Enter the user's first and last names, hit the Tab or Enter key, or click in another field to autogenerate the username, password and e-mail address values. Edit these values as necessary. Select a User Type and an E-mail Notification Level. Click on the Add button to add the user to the selected project.

For purposes of discussion, a uniquely defined W.I.N. user is someone who has a unique real name ("John Smith") and W.I.N. username ("smith") combination. Thus a W.I.N. project can contain two or more users with the same full name as long as their W.I.N. username is different.

If you select more than one project and the username you enter is already in use, you'll be asked to try another username. However, if you select a single project and someone with the same name and username already exists in that project, you'll be asked to cancel the Add request if this is the same person, or have W.I.N. generate a unique username if this is a different person.

If you're adding a user with the same last name but a different first name as an existing W.I.N. user, you may get a username conflict since the initial, autogenerated username is the user's last name. If this happens, you'll be asked if you want W.I.N. to generate a unique username for the new user, or if you want to cancel the Add request.

If the user you're adding already belongs to one or more other W.I.N. projects, their existing password and e-mail address, plus their user type and e-mail notification values, will overwrite the values entered in these fields. So don't bother editing these values when you're adding an existing W.I.N. user to a new project because they'll be overwritten by W.I.N. anyway. If you need to change one or more of these values, you may do so in a subsequent Change request.



Figure 2: Document upload form

If the user you're adding is a first-time W.I.N. user, the Add command will trigger an e-mail welcome message that explains the W.I.N. system and provides the new user with a W.I.N. username and password, which are initially the same. The user that already belongs to one or more W.I.N. projects won't receive a welcome message.

The default user type, User, can view all document versions in their assigned project, and can upload and change the status of their own documents. The Read Only user type can view all document versions but can't upload documents to W.I.N. The Field Rep user type can view final document versions only, but can't upload documents to W.I.N. or see any NonDisclosure documents identified as NDA. The Admin user type can move and delete documents; change the status of any document; and add, change and delete users and projects to W.I.N., along with other administrative functions.

If the e-mail notification level is set to "Send mail for revisions," the user is notified whenever a new or revised document is uploaded to the project. If the level is set to "Send mail for revisions and status changes," the user is also notified whenever the status of a document changes (from "Draft" to "Final" or from "Final" back to "Draft"). If the level is set to "Never send mail," the user won't receive any W.I.N.-generated e-mail except for the initial welcome message if they're a new W.I.N. user.

The Single User change command looks similar to the Single User Add.

To change a W.I.N. user's registration information:

- Select a user name.
- Click on the Single User Change button.
- Edit one or more user values.
- Click on the Change button to change the user's registration information for the selected project, or click on the Cancel button to cancel the change request.

If the user already belongs to one or more W.I.N. projects, their revised name, user-name, password and e-mail address will be copied to their other project registrations. However, any changes made to their user type and e-mail notification values will be restricted to the selected project only. Note that the user's document records will also reflect any changes made here; i.e., the user's name and e-mail address will be updated in every document record they own.

A user's project name may be changed. If it's changed to the name of an existing project to which they don't belong, they'll be added to that project and deleted from the current project. If it's changed to the name of an existing project to which they belong, they'll be deleted from the current project. If the new project name doesn't exist, the pro-

ject will be created with the user as its first member, and the user will be deleted from the current project. If the user is the only member of the current project, the effect of this change will be to delete the project itself. In all cases, whatever documents the user owns under the current project will be moved to the new project.

Select a user name and click on the Single User Delete button. Then click on the Continue button in the confirmation dialog box. If the user belongs to one or more other W.I.N. projects, you'll be asked if you want to delete them from all projects. The default is to delete them only from the project you selected.

If the user has authored one or more W.I.N. documents, you'll be asked if you want to delete the documents from W.I.N. or transfer ownership to another W.I.N. user. If you elect the latter option, you'll be asked to select the new owner from a list of all W.I.N. users. If a transferred document is in a project that the new owner doesn't belong to, W.I.N. will automatically register them for this project so they'll be able to update the document as required. The new owner's registration information will be copied from one of their existing project registrations.

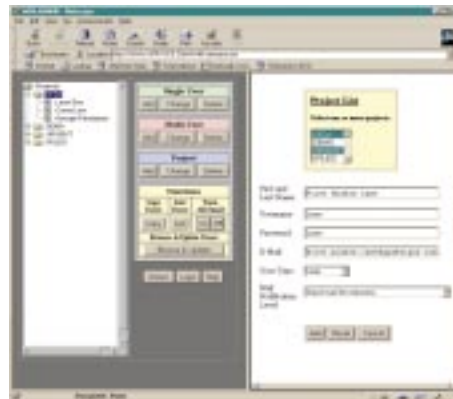


Figure 3: Single User Add

To add more than one user at a time to a W.I.N. project:

- Select a project name.
- Click on the multiuser Add button.
- Deselect the "Send Welcome" e-mail option if you don't want to send e-mail welcoming new users to W.I.N. The default is to send e-mail, primarily because this is how new users obtain their W.I.N. username and initial password.
- Type or paste one or more user names, at one per line, into the text area.
- Click on the Add Users button.

From the user's full name W.I.N. automatically generates the username, password and e-mail address. If the username already exists within the selected project, W.I.N. modifies it until it becomes unique within all of W.I.N. If this happens, W.I.N.

displays a list of all users whose usernames were modified, giving you the option of adding (the default) or skipping each individual user. Click the Submit button to either add or skip the users in this list.

If one or more users were skipped, they'll be listed in a subsequent dialog box. This dialog is informational only – no action is required other than clicking the Return button. However, you might want to print it out so you'll have a record of who wasn't added to the selected project. Up to two sections may be displayed: one for users you skipped (by clicking the "skip" checkbox in the previous dialog), the other for those who were skipped by W.I.N. because they appeared to be duplicate users (their first and last names, as well as their username, matched those for users who are already members of the selected project). In the latter case, if a duplicate user is in fact a different person, you may add them to the project with the Single User Add command described above.

To add a new project to W.I.N.:

- Select the root project name, Projects.
- Click on the Project Add button.
- Enter the new project name in the Project field.
- Enter a user's first and last name. The first user in a new project will usually be the Administrator (you).
- Hit the Tab or Enter key, or click in another field to autogenerate the username, password and e-mail address values.
- Edit these values as necessary.
- Select a User Type.
- Select an E-mail Notification Level.
- Click on the Add button to add the new project to W.I.N.
- Click on the Cancel button to cancel the add request.

To change the name of a W.I.N. project:

- Select the project name.
- Click on the Project Change button.
- Enter the new project name.
- Click on the Change Project button.

To copy users into a project:

- Select the project name.
- Click on the Copy Users button.
- Select one or more user names.
- Click on the Copy Users button to copy these users (i.e., import their registration information) into the selected project. W.I.N. won't copy a user who already exists in the selected project.

To list W.I.N. users:

- Select the root project name, Projects, to list all W.I.N. users, or select a single project name to list only its users.
- Click on the List Users List button.

- Click on the Not E-mailed radio button if you want a listing of only those users who don't require e-mail notification of document revisions or status changes.
- Click on the List Users button.
- Optionally print the user listing.
- Click on the Clear button to erase the listing.

To turn all e-mail notification either on or off:

- Select the root project name, Projects. Click on the Turn All E-mail On (or Off) button. This will turn e-mail notification either on or off for all W.I.N. users. It's recommended that you turn all e-mail off before performing test uploads or status changes within various W.I.N. projects so those users won't receive the e-mail generated by your tests.
- Use the List Users command to list users who already don't receive e-mail notifications, so you'll know whom to turn off when you turn all e-mail back on.

To facilitate database maintenance, the system is designed with a browse and update facility for the Author and Document tables. You can access the Author table by clicking the "Browse & Update" button on the administration screen. This displays the entire record, including fields reserved for future use. You can go to the first, previous, next or last record in the Author table by clicking the appropriate navigation button.

Invoking the routine for browsing and updating the Author table required a little programming finesse that combined ColdFusion and JavaScript capabilities in some interesting ways. First, we found it necessary to use JavaScript to escape the frame we were in so we'd have a full window in which to display our record data and related controls. Generally, this is accomplished by specifying a TARGET attribute with a value of "_top". However, ColdFusion's <CFLOCATION> tag doesn't support this attribute, so using this tag would confine our template output to the right-hand frame of the admin frameset. The HTML <FORM> tag supports the TARGET attribute, but we didn't want to interrupt the program flow with a form simply to obtain a larger display area. This left us with the JavaScript window location property that not only can load a URL but can also specify a target via the "top" window property (see Listing 3). Another nice aspect of this code is how we used ColdFusion to retrieve a database value (the record number for the selected user and project) and pass it to our template by pasting it into our JavaScript code using ColdFusion's <CFOUTPUT> tag. A client variable would have been another way of passing this value, but this tech-

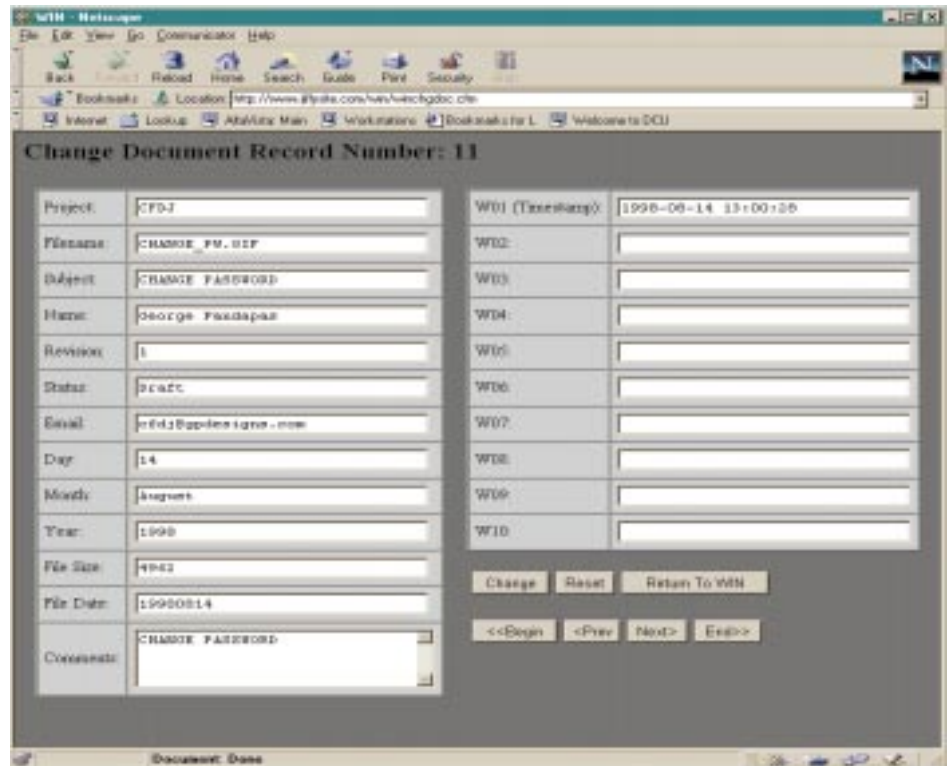


Figure 4: Browse & Update Document table

nique for dynamically generating client-side code (JavaScript) with ColdFusion can be helpful in many other ways as well.

The remaining administration functions are accessed from the spreadsheet. They are the Change, Delete and Move functions assigned to the Document table.

In the Document table (see Figure 4), as was the case for the Author table, you'll notice that this is the entire record, including 10 fields reserved for future use. You can go to the first, previous, next or last record in the table by clicking the appropriate navigation button.

The ColdFusion code for implementing these navigational controls has some interesting features that may be helpful in your own coding work. First, as with our column sort buttons earlier, the navigation buttons are submit buttons in a form that calls itself as part of the containing template, and with a hidden field value set to the current record number. Upon reentering the template, the code detects whether this is a recursive call (the hidden field parameter, "oldrec", will exist if it is) and produces a sorted list of record numbers if true. By detecting which navigation button was clicked, we can determine which record to display. For the "previous" and "next" buttons this involves decrementing or incrementing the ordinal of the previous record number to find the new record number. The final key step is to specify this value for the STARTROW attribute in the <CFOUTPUT> tag, and to be sure to limit output to only one record by setting the MAXROWS attribute to one (see Listing 4).

The administrator is the only person allowed to delete a document. This normally occurs as the result of an e-mail request from a user who has erroneously uploaded a document, or it may be that a document is simply outdated and needs to be deleted. By clicking on the Delete button for a specific document listed on the spreadsheet page, you may physically delete the record from the database.

As mentioned earlier, W.I.N. documents can be moved from project to project. New users will generally use a generic project to begin their work, and as projects are defined they will identify which documents actually belong in different projects. Transferring these documents to their designated projects is the primary reason for having the Move command. It's also used by the administrator for storing requested deletions in a holding area ("ARCHIVE") in the event that the document and the history need to be reviewed in the future.

CODE LISTING

The complete code listing for this article can be located at www.ColdFusionJournal.com

About the Author

Leon Eno, Webmaster for CVSI, Inc., has over 20 years of proactive experience in information technology. His special focus is on developing and supporting sales/marketing and financial systems. Leon also freelances in the areas of Web graphics, prepress processing and Web design, and programming. He can be reached at enol@bigfoot.com.

enol@bigfoot.com