ORNL/CP-94447

CONF-980113--

Managing Data Warehouse Metadata Using The Web: A Web-Based DBA Maintenance Tool Suite[*]

Teresa Yow, Ph.D.
Oak Ridge National Laboratory
tgy@ornl.gov

Jon Grubb
University of Tennessee
grubb@gandalf.rmt.utk.edu

Sarah Jennings
University of Tennessee
xqj@ornl.gov

Abstract

The Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center (DAAC), which is associated with NASA's Earth Observing System Data and Information System (EOSDIS), provides access to datasets used in environmental research. As a data warehouse for NASA, the ORNL DAAC archives and distributes data from NASA's ground-based field experiments. In order to manage its large and diverse data holdings, the DAAC has mined metadata that is stored in several Sybase databases. However, the task of managing the metadata itself has become such a complicated task that the DAAC has developed a Web-based Graphical User Interface (GUI) called the DBA Maintenance Tool Suite. This Web-based tool allows the DBA to maintain the DAAC's metadata databases with the click of a mouse button. This tool greatly reduces the complexities of database maintenance and facilitates the task of data delivery to the DAAC's user community.

1. Introduction

The Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center (DAAC) is one of nine data archive and distribution centers for data associated with NASA's Earth Observing System Data and Information System (EOSDIS) Project, a component of NASA's Mission to Planet Earth. The nine DAACs, located throughout the U.S., provide data support for various NASA research projects and make data available to the global change research community, policy-makers, and the general public.

The ORNL DAAC is responsible for archiving and distributing data from NASA's field experiments. Because we have multiple databases that are identical in structure but varied in content, the task of archiving, managing, and distributing this data has been a particularly challenging job. Indeed, the task of database maintenance has become so complicated that we have developed a Web-based Graphical User Interface (GUI) that allows the DAAC's DataBase Administrator (DBA) to make changes in database structure and content with the click of a mouse button. The result is a DBA Maintenance Tool Suite that integrates the friendly useability of the Web with the complexities of database management. This paper describes our problem and our customized solution. We hope it provides a useful tutorial on how you might develop such a tool to address your data management problems.

2. The Problem Described

2.1 The Databases

As a data repository for NASA's field investigations, the ORNL DAAC catalogues, archives, and distributes data to users all over the world. The data from each project archived at the DAAC is organized into "datasets," or groups of related experimental results. Each data file in a dataset is called a "granule." At the ORNL DAAC there are currently over 200 datasets, some with as many as 9000 granules. In order to efficiently archive and distribute this data, we generate metadata to describe the data and store the metadata in our Sybase® DataBase Management System (DBMS)

[*] Research sponsored by NASA under Interagency Agreement DOE No. 2013-F044-A1 under Lockheed Martin Energy Research Corp., contract DE-AC05-96OR22464 with the U.S. Department of Energy.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

tables. This metadata resides in several databases, each of which was designed to meet the specific needs of one of the three different search and order systems supported by the DAAC: there is a database for the EOSDIS Interface Management System (IMS), an X-based interface that links all nine DAACs (called the system-wide IMS for short); several databases to meet the needs of the ORNL DAAC's X-based local search interface; and a database to drive the our Web-based search and retrieval engine BIOME. The result is a system of seven metadata databases.

First, there are four scientific metadata databases that describe our data holdings. These databases are identical in structure (same tables, same column names, etc.) but varied in content since data is staged to these databases at different times during an extended system-wide EOSDIS approval process.

IngestDB is a temporary staging area where new datasets are initially entered into the DAAC's holdings after all pertinent metadata has been generated. After this metadata is successfully entered into IngestDB with no Sybase errors, it is "promoted" to the development database MetaDev, where it is tested against all of the search and extract programs. Any problems with either the metadata or the programs are resolved at this point. The metadata is then promoted to the operational database DaacMeta for access by the ORNL DAAC's two local search and access programs. At this point the new metadata is submitted for approval to EOSDIS for inclusion in the global EOSDIS IMS metadata system, which links all nine DAACs. After the new metadata is approved and ingested by EOSDIS, the metadata gets its final promotion to the DAAC's operational SystemWideOps database, which can be accessed by users of the EOSDIS IMS. After the metadata has been successfully promoted to the SystemWideOps database, IngestDB is purged in preparation for the next ingest.

There are three other databases that contain information needed by the DAAC. There are two order databases, OrderDev for developmental work and DaacOrder for operations, that contain information about users' data orders (for example, order shipping address). The WebDB is used to map certain Web operations into other tables and is used only by BIOME.

Users can browse and order data from any of the system-wide or local interfaces by specifying selection criteria including spatial or temporal coverage, geophysical parameters, or dataset names.

2.2 DBA Functions

Most database maintenance tasks at the ORNL DAAC involve manually creating and maintaining Structured Query Language (SQL) table scripts, Unix shell scripts, and Sybase text bulk copy (.bcp) files. Changes to the databases are usually one of two kinds: adding new metadata or revising existing metadata.

Adding new metadata involves first collecting metadata for all new datasets for all database tables that must be populated. The metadata is manually typed into ASCII Sybase .bcp files files with field values separated by pipe symbols (|) and missing values represented by double pipes. These files are then bulk copied into the database of choice by running Sybase's .bcp utility once for each table to be populated. Metadata is promoted to additional databases by running the bcp commands again for each table for each additional database. Adding new metadata in this manner is difficult for several reasons:

- manually typing .bcp files with pipes and double pipes invites human error;
- promoting requires remembering to run the .bcp utility for each table for each database, a process that is also error-prone; and
- maintaining shell scripts that automatically run the .bcp utility (a common DBA tactic) for every table requires remembering to modify the script any time a table is added or deleted, another maintenance problem.

Updating existing metadata is also difficult. Revisions to existing metadata generally arise following the system-wide review process, which entails scientific quality assurance and a standardization across DAACs that often results in metadata being changed several times as consensus is reached. Most of these changes affect fields in several tables across multiple databases. Revisions must be made by either

- first copying out current records using Sybase's .bcp utility, modifying the Sybase .bcp file with the new metadata values, deleting records from the affected tables in multiple databases, and bulk copying the revised file into multiple databases or
- interactively updating individual fields using SQL, a repetitive, error-prone solution at best, especially when multiple tables and databases are involved.

This process is neither efficient nor error-free. We obviously needed a better way to maintain our databases.

3. The Evolution

In an effort to automate some of the more repetitive and error-prone tasks, software developers at the ORNL DAAC developed and implemented the DBA Maintenance Tool Suite described in this paper. It is our unique, customized solution to the problem of managing the metadata for a large data repository.

3.1 Our Needs

To solve our database maintenance problems, we needed a customized tool that would perform several key functions. First, we needed a tool that would ease the repetition of making the same changes to multiple databases. Moreover, we needed a tool that would work with one, any, or all of the databases at once. Also, we had to consider the possibility of global changes in all tables, in all databases, as well as the little changes that might affect tables in some databases and not others. We needed a tool that would automate changes but allow the DBA to control exactly how and when the changes were made. We also needed a tool that would reduce the possibility of human error. Finally, because the DAAC's search and order systems have to be available on-line 24 hours a day, we needed a tool that was stable and reliable. And so the customization of our DBA tool began.

Because Sybase offers little in the way of interfaces, we needed to design a GUI interface that would "visually connect" the DBA and the databases. We chose to create the GUI on the Web for several reasons. First, a GUI Web browser would provide a user friendly interface that could reduce the tedium of database maintenance to the ease of clicking a button. Second, using the Web would allow DBA functionality from any platform running a GUI Web browser and having access to the Internet and appropriate permissions. And finally, the programmers at the ORNL DAAC have considerable experience bringing up sophisticated Web sites very quickly.

3.2 Plans for Implementation

Since ORNL had standardized on Netscape®, we made plans to take advantage of all the features of Netscape 2+. This meant using "Frames," "JavaScript," and "Cookies" whenever possible. (A Java compiler was not available for our Silicon Graphics® (SGI) Unix servers when this project started, so Java was not integrated into the design. It will be included in subsequent versions.) We would use Frames to divide the screen into a "menu" area and a "display" area. We would use Javascript for some input editing and control so that we could apply rules to the data inputs and attach convenience functions to the screens, like automatically filling in a field value with a selection from a pick list. We would use Cookies to keep track of the DBA's selections and progress.

We decided to use HyperText Markup Language (HTML) 3.2 pages and Common Gateway Interface (cgi) scripts (with some Netscape extensions). Because Sybase has a very clean and efficient interface to the C programming language, we planned to use small, specific C processes that would be executed by the cgi scripts. These C processes would access the various databases to perform database functions using Sybase's "DBLibrary" function calls. We knew such interfaces were becoming more common; what would make our DBA tool useful and robust would be the design options that we would custom build and implement.

With a list of our needs in hand and a preliminary plan for implementing the functions we wanted, the ORNL DAAC software developers, working closely with the ORNL DAAC DBA, began development of the DBA Maintenance Tool Suite, described in the following paragraphs. Although perhaps not technically revolutionary, our "needs evolutionary" Web DBA Tool is a resource every DBA deserves. Although your needs will most certainly be different, the basic approach we used should still work for you.

4. The ORNL DAAC DBA Tool Suite

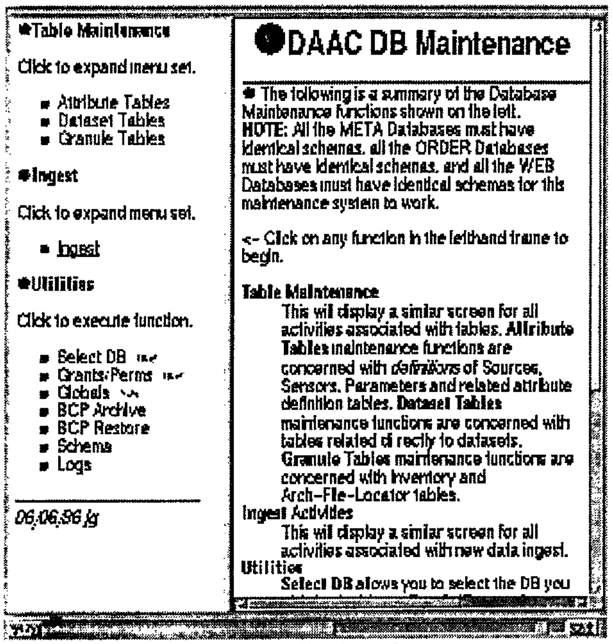
This section describes some of the more useful features of our DBA Maintenance Tool Suite. Note that the terminology and descriptions of functionality reflect the fact that our site runs on SGI Unix machines using Sybase. However, the basic philosophy should be applicable to other platforms with other Web servers and DBMSs.

4.1 Security

Our DBA Tool is protected from unauthorized access and use. Although not on a secure server, the tool does contain three levels of security. First, it uses a restricted server. To use the DBA Tool, your IP address must be in the list of acceptable IP addresses or you will get a "403 permission denied" message. Second, you must be one of a very few listed users of the tool or it won't respond to you. These users are compiled into the tool itself, and the security cannot be spoofed or bypassed. Third, you must know the DBA login and password to the database or the tool won't let you log in. Also, the access screen does not display any clear text of the access codes. These levels of security effectively control access without System Administration (SysAdmin) intervention or complex firewalls. As an added measure, the tool logs every successful DBA action it performs. The log file records what was done, when it was done, and who did it. The log files are in a private directory accessible only by the DBA tool and root.

4.2 Major Functions

After the access screen, the DBA Tool displays a screen that contains the three major categories of functionality needed for database maintenance at the DAAC: Table Maintenance, Ingest, and Utilities. Figure 1 shows these major functions.



●Table Maintenance

Click to expand menu set.

- Attribute Tables
- Dataset Tables
- Granule Tables

●Ingest

Click to expand menu set.

- Ingest

●Utilities

Click to execute function.

- Select DB
- Grants/Perms
- Globals
- BCP Archive
- BCP Restore
- Schema
- Logs

06/06/96 jg

●DAAC DB Maintenance

● The following is a summary of the Database Maintenance functions shown on the left. NOTE: All the META Databases must have identical schemas. all the ORDER Databases must have identical schemas. and all the WEB Databases must have identical schemas for this maintenance system to work.

<- Click on any function in the lefthand frame to begin.

Table Maintenance
This wil display a similar screen for all activities associated with tables. **Attribute Tables** maintenance functions are concerned with *definitions* of Sources, Sensors. Parameters and related attribute definition tables. **Dataset Tables** maintenance functions are concerned with tables related di rectly to datasets. **Granule Tables** maintenance functions are concerned with Inventory and Arch-Fle-Locator tables.
Ingest Activities
This wil display a similar screen for all activities associated with new data ingest.
Utilities
Select DB alows you to select the DB you

Figure 1: DB Tool main menu

The interface uses Frames to split the screen into a "Menu" area and a "Display" area. The Menu frame always contains either the "Main" menu or an "Expanded" menu. The Main menu contains some items that when selected, replace the menu frame with another menu that has additional selections and replaces the display frame with a help screen for those new selections. Each of the Table Maintenance and Ingest choices will expand with new menus.

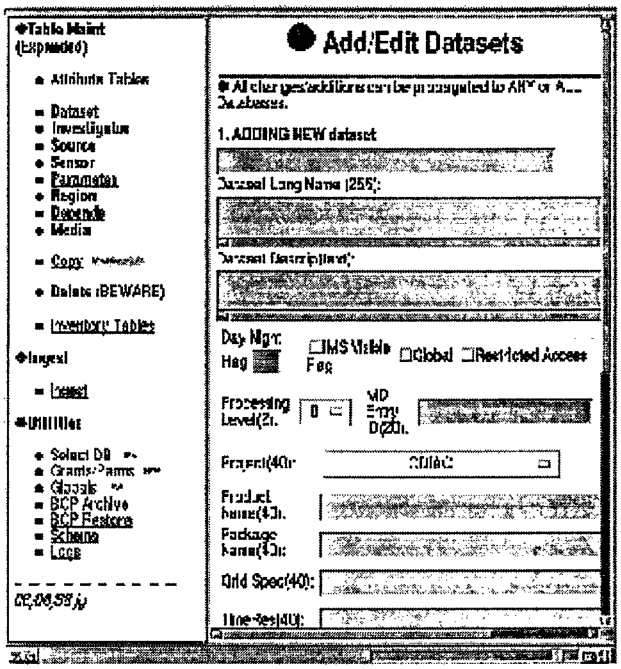
The Display area initially displays instructions on how to run the DBA Tool. It later displays progressively more detailed or selective screens. It will eventually contain the actual working screens after a menu item is selected from the menu side.

Finally, as the selected action is being executed, a "progress" screen replaces both the menu and display frames. System errors or processing errors that would violate any Unix or Sybase rules are also displayed on this screen.

The Utilities items are always on every new menu frame for easy and immediate access.

4.2.1 Table Maintenance. The Table Maintenance function contains one of the most useful features of the DBA Tool. As noted in Sect. 2.2, one of the most tedious and error-prone tasks the DAAC DBA performs is manually creating and using Sybase .bcp files to insert new metadata or update existing metadata. The DBA Tool solves this problem by providing template screens that mirror the structure of the table to be updated.

Let us say that we want to modify the dataset table. We click on Dataset Tables under the Table Maintenance Menu (see Fig. 1) and are offered the option (not shown here) of either updating an existing dataset record or adding a new record. Then we see the dataset template (Fig. 2), which is the same for either adding or updating except that when new datasets are being added, the data fields will be blank.



●Table Maint (Expanded)

- Attribute Tables
- Dataset
- Investigator
- Source
- Sensor
- Parameter
- Region
- Dependb
- Media
- Copy
- Delete (BEWARE)
- Inventory Tables

●Ingest

- Ingest

●Utilities

- Select DB
- Grants/Perms
- Globals
- BCP Archive
- BCP Restore
- Schema
- Logs

02/06/98 jg

● Add/Edit Datasets

● All changes/additions can be propagated to ANY or ALL Databases.

1. ADDING NEW dataset

Dataset Long Name (255):

Dataset Description(;

Day Night Flag ☐IMS Visible Flag ☐Global ☐Restricted Access

Processing Level(2): 0 MD Entry ID(20):

Project(40): CDIAC

Product Name(40):

Package Name(40):

Grid Spec(40):

Time Res(40):

Figure 2: DBA Tool template screen

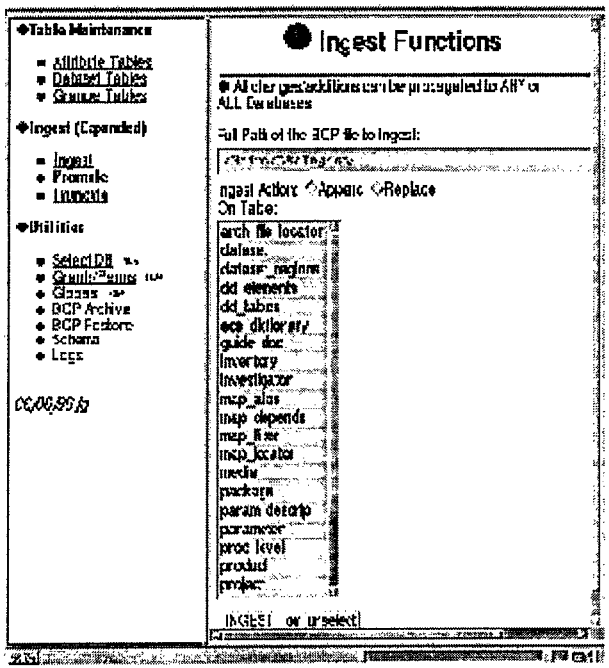
If we select an existing dataset, the screen items are completed with the current dataset values for those items. Any item except the dataset name (the primary key of the record) can be altered.

The template screen is programmed such that appropriate Sybase rules (e.g., field length, data type) are applied for the fields that are being input. The screens employ JavaScript to ensure correct combinations of information. Mandatory fields must be supplied before the table update is actually performed. The program also checks to be sure that fields like latitude and longitude are valid values. For a few of the fields, there are pulldown menus listing all the possible correct entries.

After all fields are completed, we can either choose to append the record to a .bcp file to be copied later into the dataset table (see discussion of Ingest in Sect. 4.2.2), or we can choose to commit the update transaction to the selected databases immediately.

4.2.2 Ingest Functions. Another powerful and useful feature of the DBA Tool is the Ingest function set. As described in Sect. 2.2, ingest involves generating a complete set of metadata tables for new scientific datasets that are being added to the DAAC's data holdings. The usual procedure is to manually create a complete suite of ASCII .bcp files for bulk copy, one by one, into the database. Obviously, this method leaves room for error, particularly since the data has to be promoted up through multiple databases.

The DBA Tool solves this problem. To ingest new metadata, we first select Ingest from the main menu (Fig. 1) to display the expanded Ingest screen (not shown), which displays three options: Ingest, Promote, and Truncate. First we click on Truncate to delete all old metadata from IngestDB (IngestDB always contains only new data that has been ingested; it is purged completely after each data promotion). Next we select Ingest to bring up the main Ingest screen, seen in Fig. 3.



● All changes/additions can be propagated to ANY or ALL Databases

Full Path of the BCP file to ingest:

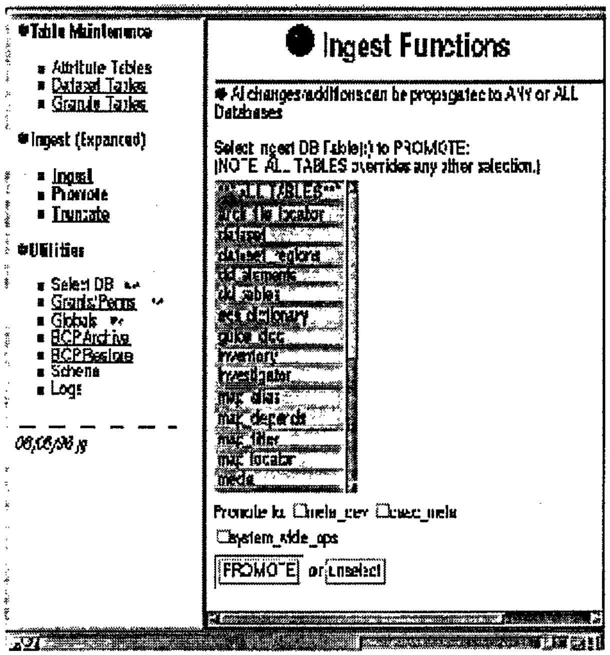
Figure 3: DBA Tool main ingest screen

To ingest, we first input the complete path of the desired .bcp file created with the templates described in Sect. 4.2.1. Then we select which table this .bcp file is to be copied to.

With just a few clicks, data can be moved into IngestDB without the DBA having to issue one Sybase command or run one shell script. This process is repeated with each table to be populated until all of the new metadata is in IngestDB.

An even more critical part of Ingest is the ability to promote from one database to another. Promotion from database to database is possible because, as described earlier, the metadata databases are identical in structure; that is, they all have the same tables, the same columns in tables, etc. So a promote is a rather straightforward append from one table to an identical table in the next database, and so on until all tables have been promoted.

As can be seen in Fig. 4, at the DAAC we are able to promote new data from the IngestDB to any or all of the other databases with only a couple of clicks.



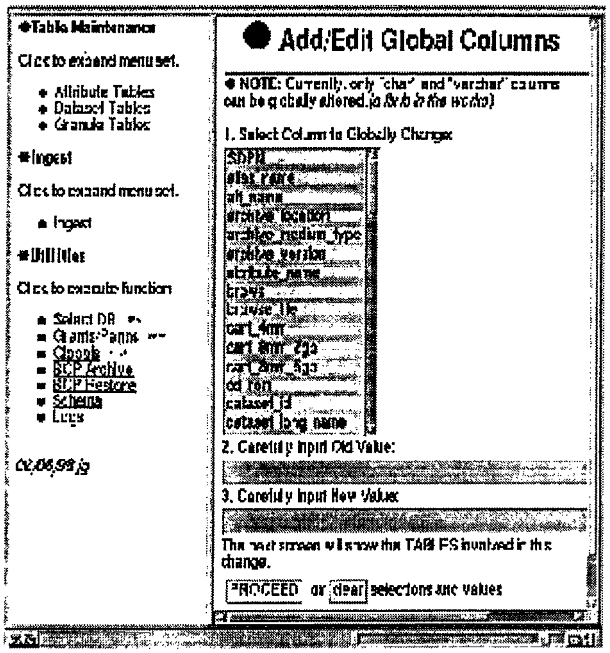
● All changes/additions can be propagated to ANY or ALL Databases

Select Ingest DB Table(s) to PROMOTE:
(NOTE: ALL TABLES overrides any other selection.)

Figure 4: DBA Tool promote function

To do this, we select which tables to promote from IngestDB (ALL TABLES is the usual choice for promotion), and then we indicate which database to promote to. We always promote to MetaDev first because it is the development database. Then we move to DaacMeta, the operational database, and finally to SystemWideOps, the final resting place of all data officially sanctioned by EOSDIS and NASA. The DBA TOOL ensures that the DBA is always in control of the timing of database promotions and that the databases can be brought into sync quickly and easily.

4.2.3 Utilities. The Utilities features of the DBA Tool are made up of highly customized tasks that are performed often in the course of database maintenance. Figure 5 shows one of the most useful of these functions: global updates to the databases. Let's say the NASA system-wide approval process described in Sects. 2.1 and 2.2 has determined that the field value "CO2" must change to "Carbon Dioxide." This means this particular value must change not only in every table in which it appears but also in every database in which these tables appear. Although the ORNL DAAC has attempted to achieve third-normal design, some fields necessarily appear in more than one place. Updating this field in all tables and in all databases poses a DBA nightmare. Luckily, the DBA Tool provides a global update function (Fig. 5).



Add/Edit Global Columns

NOTE: Currently, only "char" and "varchar" columns can be globally altered. [illegible]

1. Select Column to Globally Change:

2. Carefully Input Old Value:

3. Carefully Input New Value:

The next screen will show the TABLES involved in this change.

PROCEED or clear selections and values

Figure 5: DBA Tool global utility

To use this feature, we simply select the field to update, input the old value and the new value, and then select the tables and databases in which to make the changes. With a few clicks all occurrences of a particular value can be changed, quickly and easily.

Other Utilities functions include the SelectDB option, which allows the DBA to select which database to use as the starting or "current working" database; the "Schema" screen, which allows the DBA to create table schemas for any and all tables in any and all databases; and the "Logs" option, which allows the DBA to display the log files kept on all DBA actions on the database. The tool keeps log files by the month so the DBA can go back and select any particular month. This technique also keeps the log files from being unmanageable on the screen. (They overwrite themselves annually so there is always 11+ months of logs.)

5. Conclusion

We do not claim to have created a Web tool that is on the cutting edge of Web technology. As a matter of fact, all of the functions except the menus and help screens use Unix Bourne Shell scripts and C code modules to perform the required actions. The elegance is not in the implementation but in the design. Our DBA Maintenance Tool Suite does exactly what is needed by the DBA, nothing more, nothing less. It's like magic.

You can do the same magic for your operation. The magic is to meld your needs with your site's ability to build a custom DBA tool. Remember that you may have to modify your system somewhat like we did to make the tool more useable (for instance, we had to modify our databases to make sure they were identical so operations would work the same on all databases). Your tool will also reflect the access mechanisms of your DBMS (our Sybase has a very nice C interface) and the capabilities of your site's standard browser (Netscape has very useful extensions). For a very useful and robust DBA tool, use the very best Web programmers you can afford. In tools, quality counts more than anything else. After all, how much will you use a tool that is unreliable?

We have demonstrated that custom Web-based DBA tools can be successfully deployed. Every DBA deserves a great DBA Tool Suite.

Biographical Information

Dr. Teresa Yow, who has a Ph.D. from the University of Tennessee in Knoxville, is a Systems Analyst in the Computational Physics and Engineering Division of the Oak Ridge National Laboratory in Oak Ridge, Tennessee. She is database designer and database administrator for the ORNL DAAC.

Mr. Jon Grubb is a Senior Systems Designer and Integrator for the Pellissippi Research Center of the University of Tennessee at Knoxville, Tennessee. He designed and implemented both the NASA System-Wide Server and the BIOME Web Server for the ORNL DAAC.