

CMB Database

ColdFusion Implementation

June, 1999

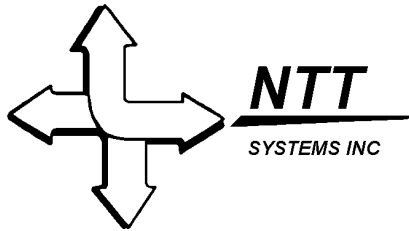
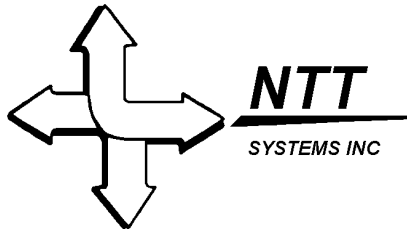


Table of Contents

1.	Background	1
2.	Architecture.....	1
3.	ColdFusion Setup.....	2
4.	Browser Considerations	2
5.	Security, User Groups and Policies.....	2
5.1	Directories and User Groups.....	3
5.2	Rules	3
5.3	Policies.....	3
6.	Forms, Reports, and Printing.....	4
7.	Design Details	4
7.1	Data Entry Forms.....	5
7.1.1	Populating Form Fields and Record Navigation.....	5
7.1.2	Adding, Editing and Deleting Records.....	6
7.2	Composite Reports.....	6
7.3	Printed Reports	6
7.4	Files that Comprise the Application	6



1. Background

The CMB medical database is used to hold information relating to CF personnel who come to CMB as part of an ASC, as well as those who are referred for other medical purposes. The database has a long history and has been implemented on several platforms. Most recently the configuration consisted of an ORACLE 6 back end on a SUN server with MAC clients running an OracleCard application.

All of the hardware and software that comprise this system was declared to be non-Y2K compliant. The replacement system, documented here, was developed to remedy this situation. The goals for the project were:

1. To build a Y2K compliant replacement for the then current system.
2. To maintain all of the functionality of the current system.
3. Where possible, to address some of the deficiencies of the current system.
4. To build the replacement system in a manner that would minimize retraining requirements.

The system, delivered in June 1999, met all of these goals. Third party software (ColdFusion and Oracle 7) is certified Y2K compliant by the vendors and the application code is certified compliant by NTT. DCIEM is responsible for ensuring the Y2K status of other components (hardware, operating systems, and related software).

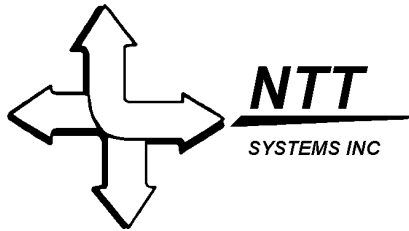
The new system uses the current database (structure and content), and the new forms are closely modeled on the originals. This maintains functionality and minimizes retraining. The new implementation addresses the most complained about features of the original. In particular, records can no longer be inadvertently be modified or added.

The following documentation describes the structure and design principles that underlie the new CMB database application.

2. Architecture

The system is implemented as a standard, three-tier client-server application. The Oracle 7.3 DBMS sits at the top. It runs on a SUN server running UNIX.

The middle tier consists of a web server and the ColdFusion application server. They run together on an Intel based NT server. The application demands very little of the web server; simply the ability to serve HTML documents to clients, and to pass .cfm requests to ColdFusion for processing. Microsoft IIS is used as the server for the CMB application. We chose the "Professional" version of ColdFusion since the more advanced features of the "Enterprise" version (load balancing, fail over and native database drivers) were not required.



Web browsers implement the bottom tier. One of the reasons for selecting ColdFusion for the application was that sends standard HTML to the browsers. Any additional feature requirements are the result of explicit design choices made by the developers. We chose to minimize these and the application will run on any browser that supports HTML 3, style sheets, and JavaScript 1.2. We have tested the application successfully with Netscape 4.0 and IE 4.0. It is optimized for IE (the DND standard) and certain usability features (eg., input box background colour) are not available from Netscape.

3. ColdFusion Setup

ColdFusion installs itself as a service on NT systems. We performed an “out of the box” installation without any customization. The *Advanced Security* features must be enabled since we use user account and group information to control access to features and the database. All caching is disabled. See Section 5 below.

4. Browser Considerations

Any browser that supports HTML 3, CSS-1, and JavaScript 1.2 can be used as a client for this application. Certain features, such as field highlighting, will currently only work with IE 4.0 and above. Specifically, they do not function with Netscape 4.0.

NOTE that for ALL browsers, preferences must be set up so that the browser will not use a cached document without first checking with the server. For IE 4, this is done from the *Internet Options* menu item. Use the *Settings* button on the *Temporary Internet Files* section of the *General* tab and select *Every visit to the page*. **Failure to do this will result in the browser not displaying database updates and additions.**

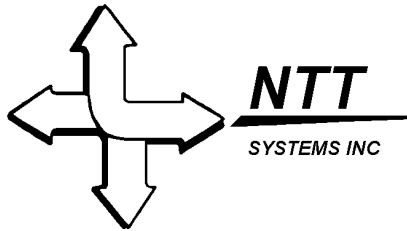
Forms are designed assuming a 1024x768 screen resolution.

Browsers must have access to a printer definition that locates the physical printer that will be used and also specifies 80% scaling.

5. Security, User Groups and Policies

Security is implemented at two levels, external and internal. The external security system controls access to the application and it is implemented by the corporate firewall and the CMB web server (the latter by specification of which IP addresses the server will respond to).

The internal security system is implemented cooperatively by the ColdFusion server and application software. The application can ask ColdFusion to authenticate a user’s name and password, and this is done using NT user directories. Once this is done, we know that the user is a known person and the question now is; what functions are they allowed to perform? This is resolved using directories and ColdFusion *rules* and *policies*. ColdFusion’s security system is described in their documentation; *Administering ColdFusion Server; Ch. 9, Advanced Security*, and *Advanced ColdFusion Development*;



Ch 14, Application Security. The application's security scheme is defined completely in the *aircrew_Box* security context. It defines, or uses, the following directories, rules, and policies.

5.1 Directories and User Groups

CMB Database users currently can all be authenticated through a single user directory. The name, *cmb_domainUsers*, is defined through ColdFusion Advanced security and points to *NT_Server* as the directory of name/password pairs. Anyone who wants to use the application must belong to one of the following groups:

- *cmbguest*
- *cmbuser*
- *cmbmanager*
- *cmbadmin*

5.2 Rules

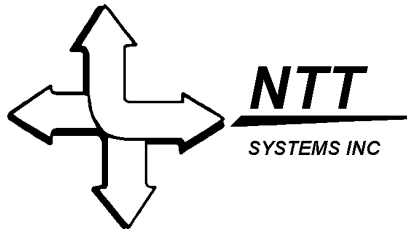
ColdFusion's use of the term *rule* is a little odd. A rule is in fact an application object defined on the server. A rule can be associated with user groups or individual users to create a *policy*. The following rules were created in the *aircrew_Box* security context.

- *InaircrewApp* represents the ability to use the application.
- *PersonalRule_Obj* represents the ability to create new records in the personal data table.
- *LetUserIn* represents the ability select data from the database.
- *EditRule_Obj* represents the ability to edit and delete data.

5.3 Policies

Policies combine users/groups and rules in a way that means *let these users do this*. The application creates a hierarchical set of policies as follows.

- *personalAccessLev* – Allow the user to create new personal data records.
rules: *PersonalRule_Obj*,
users/groups: *cmbuser*,
- *DatAccessLev1* – Allow the user to select and display records.
rules: *LetUserIn*, *InaircrewApp*,
users/groups: *cmbadmin*, *cmbguest*, *cmbmanager*, *cmbuser*,
- *DatAccessLev2* – Allow the user to delete recent records and edit all records.
rules: *EditRule_Obj*,
users/groups: *cmbmanager*, *cmbadmin*



6. Forms, Reports, and Printing

The application presents two types of forms to the user; personal data entry, medical data entry. The former is used to create entries for new individuals and is used to enter demographic data such as name, rank, service number, address, date of birth, etc. The second is used to enter specific medical data, and a unique form is provided of ecg, eeg, vision, hearing, etc. The layout of all of these forms is based on the OracleCard implementation of the database. Specific fields from the personal data form are presented as non-editable text at the top of each form.

In addition, all forms are displayed with a form navigation drop down control. Users can move between forms (for the same individual) by selecting the desired form and pressing the *Go* button. *DataAccessLev2* users have an additional control that allows them to add, edit and delete data records. Note that only records no more than fourteen days old can be deleted.

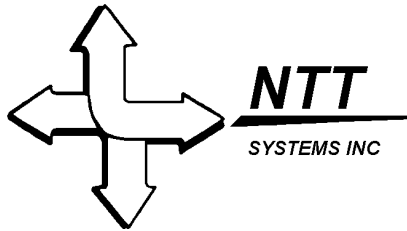
The system produces the following forms in a non-editable format suitable for printing and filing or sending to external agencies. These are as follows:

- *Flight Surgeon's Report* – A formatted selection of data from the personal, anthro, vision, hearing, and flight surgeon tables.
- *BI Report* – A concatenation of the initial medical, vision, hearing, chest X ray, ecg and lab data forms
- *B2 Report* – A concatenation of the anthro, pulmonary, echo, eeg and flight surgeon forms.
- *Echo Report* – A reformatted version of the echo form suitable for sending to outside agencies.
- *Diagnostic Echo Report* – A reformatted version of the diagnostic echo form suitable for sending to outside agencies.

All forms and reports can be printed by selecting *Print* from the browser's *File* menu. The size and print quality of the output is solely up to the browser and underlying operating system. *The application exerts no direct control over printing.*

7. Design Details

Users enter the system by specifying <http://...../index.cfm> as the url. This will take them to the splash page that allows them to select individuals and enter, edit and view data. Each time the user accesses this location from a new browser session or after one hour of inactivity, they will be challenged for their user name and password before the splash screen is presented. Successful login creates a new session for the user along with a set of set of session variables that keep track of user state information.



When the user requests any *.cfm*, ColdFusion first executes *Application.cfm*, and this module determines whether or not a session is in progress. If so, control passes to the *.cfm* referenced in the post. If not, the login screen is sent to the user. The login attempt is detected and validated when the user responds.

7.1 Data Entry Forms

All data entry *.cfm*'s construct their forms in the same way. First, *CMBHeaderGrpA.cfm* or *CMBHeaderGrpB.cfm* is processed, depending on whether or not the user has editing capabilities. The two templates are very similar, differing primarily in whether or not the editing menu is presented and in the inclusion or exclusion of validation logic. The header template is responsible for performing the SQL select statement that extracts a record set from the relevant table. Multiple records can be returned, since an individual may have multiple vision records, for example. The record set is converted into a ColdFusion WDDX structure and sent as a JavaScript array to the browser.

The header then generates a “<script src=“controlScripts.js” language=“JavaScript”></script>” that causes the browser to request the file containing the toolkit of function written for the application. These consist primarily of browser management, and record display and navigation code.

Next, other JavaScript support routines are generated, some of which may override functions first defined in *controlScripts.js*. This is done if special processing is required depending on user privileges.

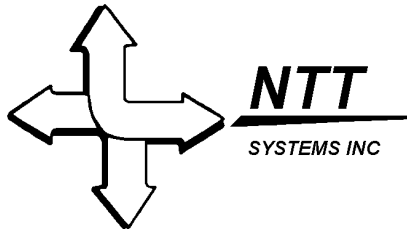
Finally, the <Body ...> and <Form ...> segments are generated. Two things should be noted. First, the <BODY> tag has an *onLoad* event that populates the form controls with data from the newest record in the record set. Second, the form is started by the header *.cfm* but completed in the *.cfm* that was requested. This is done because the beginning of all data entry forms is the same and it is generated from a combination of session data (captured from the personal information file when the individual was selected), and common information from the medical table that was requested.

The remainder of the form is generated from the target *.cfm*, eg., *labdat.cfm*, *ecgdat.cfm*, etc.

7.1.1 Populating Form Fields and Record Navigation

Each time the user requests a form, the application sends back a combination of HTML and JavaScript. The HTML creates an empty form and the JavaScript includes the record data. The <BODY> tag triggers an *onLoad* event that calls our *loadData* routine. This in turn uses the newest record to populate the fields of the form. *loadData* relies on the convention that data fields on the form have the same names as the database table columns that they are to display. The algorithm simply checks each column name in the record set, and if there is a field in the form with the same name, the record contents are transferred to the form field.

If multiple records have been returned the user can navigate between them using the “<” and “>” buttons on the form. This causes *loadData* to display the previous or following record as just described. This approach avoids interaction with the server.



7.1.2 Adding, Editing and Deleting Records

Users with sufficient privilege can add, edit and delete records. The “delete” action can only be applied to records less than fourteen days old. This was done to allow for dealing with erroneous data entry while at the same time providing for long term data stability.

A drop down box provides access to these functions. Data are initially presented read only. The input fields will refuse to accept focus. If the user selects *Add*, a green background will be applied to all input fields and field content will be cleared or set to appropriate defaults. Selecting *Edit* will not modify field content, but background colour will be set to yellow.

The user completes the action by clicking a *SUBMIT* button. This first validates the input using the function *formValidated* and if successful, data is sent back to the server for insertion into the database. *formValidated* is a common routine that performs validation based on a list passed to it. The list is private to each form and consists of triplets; field name, data type specific validation routine, and error message text.

The user will get a warning if they attempt to leave without first submitting the update or addition.

7.2 Composite Reports

There are two composite reports that are used by medical staff to evaluate individuals that come to CMB in the ASC program. These are known as B1 and B2. These are constructed by simply concatenating the forms that comprise them. Data in the reports can not be edited.

The SQL select and data loading logic differs from that described above. For these reports, only the most recent record from each required table is selected and *loadData* has to be modified to take multiple forms and record sets into account. These requirements are implemented in *CMBHeaderB1B2.cfm* that is used in stead of *CMBHeaderGrpA* or *CMBHeaderGrpB*.

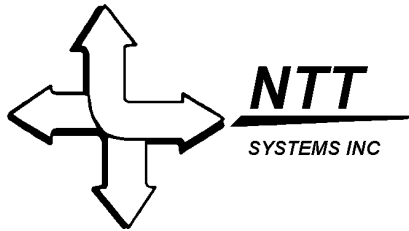
7.3 Printed Reports

Since the browser is used for printing, frames are printed exactly as they appear. This is a problem in cases where printed forms are used for filing or transmission to outside agencies since they require different formatting. We deal with this by providing “print format” versions of the forms. This is done for echo, diagnostic echo and flight surgeon forms. The printed versions can be displayed by pressing a “view report” button. Once the new form is visible it can be printed with the browser’s *print* function.

7.4 Files that Comprise the Application

The application is delivered as a main directory containing all application *.cfm* and JavaScript files, and two subdirectories, one for logos, and one for the help files. The application files can be categorized as follows:

- Application Main



Application.cfm – is called by the ColdFusion server as a prolog before executing any other *.cfm* file. Used for user authentication and log in.

- Header Files

CMBHeaderGrpB.cfm – A header included for all input forms for users who have *basic* privileges (no edit, add or delete).

CMBHeaderGrpA.cfm – A header included for all input forms for users who have *advanced* privileges (edit, add, delete).

CMBHeaderB1B2.cfm – The header for the B1 and B2 composite reports.

CMBHeaderRpt.cfm – The header for all print format report forms.

- Form and Report Files

xxxDat.cfm – Data entry forms. Named after the corresponding database table, *eg.*, vision table → *visiondat.cfm*, ecg table → *ecgdat.cfm*.

b1Dat.cfm, *b2Dat.cfm* – B1 and B2 composite reports.

xxxxxxRpt.cfm – Print format report. Named after the corresponding data entry form, *eg.*, *ecgDat.cfm* → *ecgDatRpt.cfm*.

- JavaScript

controlScripts.js – Toolkit of functions used by all forms. Some functions may be overridden in form specific *.cfm*'s.

ValidateData.js – The main validation routine and a number of functions that can be used for specific types of validation, *eg.*, numeric, date, non-empty, *etc.*

wddx.js – Provided by Allaire. A set of JavaScript functions used to serialize and deserialize record sets for transmission between the ColdFusion server and JavaScript running on the browser.

- Style Sheets

styleSheet.css – Hypertext style sheet used for the layout of all forms and reports.

indexSheet.css – Hypertext style sheet used for the help system.