

**Requirements Document
For
The Sharing of Meta-Data Between
the IHFS DB and the RAX Archive DB**

April 8, 2005, revised June 15, 2005

1 Introduction

This document is the first step of the RFC Archive Database Update and Maintenance (RAXUM) team in developing a set of requirements describing the functionality of proposed new applications for the sharing of Meta-Data between the Integrated Hydrologic Forecast System (IHFS) database (DB) and the RFC Archive System (RAX) Archive DB. The focus is on this process because more than one River Forecast Center (RFC) has mentioned having difficulty with the currently available tools to keep the RAX Archive DB in-sync with the IHFS DB for a variety of Meta-Data. For the purpose of this document, the term "Meta-Data" refers to information that describes a specific location, such as station name, latitude and longitude of the location, county and state of the location, stage-discharge relationship for this location, etc.

Feedback from various RFCs over the past several months has provided the following generalized statements about applications involved in the Meta-Data sharing process.

Need a RAX GUI application to access the meta-data and quality control tables. Something like a stripped down version of the WHFS Hydrobase application. Application needs to some how see selected tables in the IHFS database (i.e., location, ingestfilter, rivercrit, etc) (ADB ER-12, Archiver Bug List on RFC Field Support webpage).

From telephone conversations: an application that runs in batch, either manually or as a cron job, that transfers information from IHFS database location and ingestfilter tables to the adb location and ingestfilter.

Both of these descriptions are way too broad, and could be interpreted in any number of different ways by a programmer. In early February 2005, the RAXUM Team Leader sent out a request to the 13 River Forecast Centers requesting input to help define the requirements for the sharing of Meta-Data between the IHFS DB and RAX Archive DB. Eight responses were returned by the due date

of March 4, 2005. This document attempts to better define the application requirements for this data sharing process by summarizing the RFC feedback.

1.1 Currently Available RAX Features and Tools

The RAX Archive DB currently allows for a history of information on Meta-Data to be stored in the following tables: location, rivercrit, reservoir, rating, ratingshift, crest, slopeprofile and adjustfactor.

The adbinit application suite (locatdef, ingestdef and some scripts) allows the user to initialize the location and ingestfilter tables when the system is first setup. While not intended for repeated use, the adbinit suite can be used to add new entries to both tables.

All tables in the RAX Archive DB have ISQL forms, which can be used to add, modify or delete data.

No RAX baseline applications, OB1 thru OB4, are currently available that will allow for initialization or updates to the tables reservoir, rating, ratingshift, crest and slopeprofile.

However, MBRFC did write a local application to create load files for initializing the rivercrit, rating and ratingshift tables. Their application was specific to their operations. The application was provided on request to other RFCs with the caveat that they would probably have to do some tweaking of the code before the application could meet their needs.

As part of the build OB5 *After Install Procedures* (Part 13, step 4. b.), an application, cr_rvrcri, is provided. This application allows a RFC to initialize the rivercrit table. User documentation for this application is available on the RAX on-line documentation web page.

1.2 General Requirements

Meta-Data transfer from the IHFS DB to the RAX Archive DB should meet the following general requirements:

- 1) Not all IHFS DB tables are needed on the RAX Archive DB. Based on a thorough review of the existing tables, those necessary for archival should be the only ones mirrored on the RAX.
- 2) A user must be able to track changes so they can know what was in the database at a specific time in the past. For example, all station lat/lon changes should be logged and a user should be able to review those logs. Another example is rating curves, changes in the rating curve need to be tracked, so that accurate conversion of

stage to discharge in the RAX Archive DB can be made. This tracking may only be needed for a subset of tables. This should be determined after more thought and study.

3) Changes should move from the IHFS DB to the RAX Archive DB automatically and be transparent to the user. This could be done each day, or with some sort of trigger. The changes could be propagated to the archive when they are made. There should also be a way to manually push any changes from the IHFS DB to the RAX Archive DB.

4) As a general rule, any corrections needed to the RAX Archive DB should first be made in the IHFS DB and these changes should then be passed to RAX so the RAX Archive DB is updated.

1.3 PostgreSQL

The RAXUM team, in coordination with OHD/HL, is working on the migration of the Archive DB from Informix to PostgreSQL as a part of build OB7. The IHFS DB moves to PostgreSQL in build OB6. Most of the applications mentioned in section 1.1 that access the IHFS DB will no longer be available as of OB6. With the implementation of PostgreSQL for the Archive DB in build OB7, some of the other previously mentioned methods of updating selected meta-data tables may also be lost. The RAXUM team is exploring some possible replacements. For example, pgforms may be a replacement for ISQL forms. In other cases, the RAXUM team may need to recruit someone to write an application to replace a function that exists in Informix but is not in PostgreSQL.

These proposed applications will be accessing the IHFS and Archive databases in the PostgreSQL environment.

2 IHFS-side Process

Several tables in the IHFS DB contain Meta-Data and Quality Control information, which can also exist in the RAX Archive DB. The following chart indicates the IHFS DB table(s) and its corresponding RAX Archive DB table(s) in which some or all of the data resides.

IHFS DB Table	RAX Archive DB Table
location	location
ingestfilter	ingestfilter
riverstat, floodcat	rivercrit
rating	rating
ratingshift	ratingshift
crest	crest

reservoir	reservoir
datlimits	datlimits
locdatlimits	locdatlimits
adjustfactor	adjustfactor

While the tables in the two databases may have the same names, the table structures are not identical. Many tables in the RAX Archive DB are designed to allow for a history to be maintained. In other words, multiple rows for the same location identifier are allowed and the table's primary key generally includes a date.

There are several methods that can be used to add/modify data in the Meta-Data tables in the IHFS DB, including: WHFS Hydrobase, Isql forms, load files, and sql update/insert queries. While WHFS Hydrobase is the preferred way to add/modify information, it must be recognized that large, wholesale changes might also be made to the IHFS DB using SQL, load files, etc. Therefore, the updating application needs to remain independent of the WHFS Hydrobase application.

The IHFS-side application needs to have the following capabilities:

- 1) A method, on the IHFS-side, needs to exist that can capture when changes are made to the aforementioned tables. One way to accomplish this might be thru the use of triggers in the IHFS DB. Whatever the method decided on, this process needs to create "log files" for each table that include all of the information needed by the RAX Archive DB.
- 2) These "log files" should be automatically passed to the RAX.
- 3) It is recommended that these log files should be a standard delimited format. Delimited format files provide a convenient "paper trail".
- 4) The application needs to allow the user to specify which of the IHFS DB tables are to have changes tracked (i.e. logged). This user selection feature would exist for these tables: location, ingestfilter, riverstat, floodcat, rating, ratingshift, crest, locdatlimits and adjustfactor
- 5) The IHFS-side application must have the capability to manually create/append to the log files all changes in the selected table(s) since a specified date.

Another approach might be to:

- 1) Dump the IHFS tables into files which reflect the table structure on the RAX.
- 2) Read these files and compare them to the files on the RAX, updating only the rows found to be different.
- 3) Log only the updated tables and rows.

- 4) This would run on a cron. Triggers will not ensure consistency between systems. A batch approach could be run anytime and consistency would be maintained.

The approach to be used will be determined at actual development time.

3 RAX-side Process

Based on the eight responses received, most RFCs feel there should be two applications on the RAX. One that is a graphical user interface (GUI) modeled after the WHFS Hydrobase application, and one that is a batch application that can be run manually or as a cron job. Section 3.1 describes the functionality of the GUI and section 3.2 describes the functionality of the batch application.

3.1 GUI Application “arcbase”

The RAX GUI application should:

- 1) be similar in appearance to WHFS Hydrobase application. (Additional information on the WHFS Hydrobase application can be found on the WHFS Field Support Group webpage.)
- 2) have no password protection to access the GUI
- 3) pop-up an empty window and prompt the user for a location identifier (LID), instead of providing an initial listing of location table information. Once a LID is entered, the GUI should show all location table entries for that LID
- 4) wherever possible, have similar names and functions on the GUI features (such as the pull-down menus).
- 5) have the following pull down menus:

File -> preferences, exit

Location -> add location, modify location

RiverGage -> River Gage, Rating Info, Crest History, slopeprofile, avg,

Reservoir -> reservoir

Data Ingest -> Ingestfilter, Adjustfactor, QC data limits, sensok

Reference -> country, state, counties, huc2, huc4, huc6, huc8, wfo_hsa, rfc, shefdur, shefex, shefpetrans, shefpe, shefpe1, shefprob, shefqc, shefts, agency, prod

NWSRFS -> modctrl

ProcessIHFS -> location, ingestfilter, adjustfactor, rivercrit, crest, rating, ratingsshift, reservoir

Pull-down menus **File**, **Location**, **RiverGage**, **Reservoir**, and **DataIngest** should have options and functionality similar to the WHFS Hydrobase pull-down menus with the same name, although there will be some differences. One difference would be in the editing option for the ingestfilter table, the dur and idur fields should be linked so if one field is changed, the other field would automatically be populated with the corresponding value. Other differences are to be decided later.

The **Reference** pull-down menu replaces the Admin menu that is in WHFS Hydrobase.

Pull-down menu **NWSRFS** is new and would allow access to the modctrl table.

The **ProcessIHFS** menu is new. The user would select one of the tables listed as an option in the pull-down menu and then the application would find the log file of IHFS DB changes for that table. The application should read an entry and display that information along with the most current entry already in the selected table, if one exists. If an entry exists, it should highlight what has changed and prompt the user to “skip” or “accept” each change. A new entry would be generated based on what the user accepted. Fields that did not have changes would automatically migrate to a new entry. If no entry exists in the RAX Archive DB, then it should prompt the user to “skip” or “accept” the entire set of data. In the case of the location table, if an entry already exists, the sed file of the existing entry would be set to the date the log file was created, and the new entry would have its sbd field set to the same date, while the sed field would be set to null.

3.2 Batch Application

The RAX batch process should:

- 1) Allow the user to define which tables will be updated.
- 2) Allow the user to run the process manually or as a cron job.
- 3) Read the log files generated by the IHFS-side process; and check to see if an entry already exists. If it does, depending on the table, the batch process should either insert a new entry or update information in an existing entry.
- 4) Generate a log of changes made to the RAX Archive DB by this application for review.