E-2. run_raxdb_sync (IHFS DB -> RAX DB Synchronization)

1.0 General Information

1.1 Application Description

The Integrated Hydrologic Forecast System database (IHFS DB) - RFC Archive database (RAX DB) synchronization application is written in Java and consists of the following:

run_raxdb_sync (Korn script)
rax_apps.jar

1.2 Enhancements/Bug Fixes/Changes

Build OB8.3

This application is new in build ob8.3

1.3 Design Considerations

This application is one of two developed under the HOSIP project NID-06-010 - SON-06-001, *"Synchronize RFC Archive Database & IHFS Database Metadata".*

Selected meta-data is synchronized between the two databases. Keeping the meta-data contained in these databases in sync is vital for effective forecasting, research and verification. Changes are one directional, changes are made ***only*** to the RAX DB. The table below indicates the IHFS DB table(s) and the corresponding RAX DB table in which some or all of the data reside.

IHFS DB Table	RAX DB Table
location	location
ingestfilter	ingestfilter
riverstat, floodcat	rivercrit
rating	rating
ratingshift	ratingshift
crest	crest
reservoir	reservoir
datalimits	datalimits
locdatalimits	locdatalimits
adjustfactor	adjustfactor

Table 1. IHFS DB to RAX db table mapping

An integral difference between the tables of the IHFS and RAX databases is that a history of meta-data changes can be accumulated in most of the RAX DB meta-data tables.

It needs to be noted that the table structures in the two databases are not identical. All data and text case conversions will be done within the application. The databases (IHFS DB and RAX DB) to be synchronized must reside on the same AWIPS system and must be running.

1.4 Assumptions the application makes

It is assumed that changes in meta-data are made to the IHFS DB, primarily through the WHFS Hydrobase application. However, it must be recognized that large, wholesale changes might also be made to the IHFS database using SQL, load files, etc. These changes are then migrated to the RAX DB via the synchronization software.

For each of the ten RAX DB tables mentioned in the previous section, there are 3 sets of rules, these are:

1) rules for inserting a new database record
2) rules for determining if a database record is synchronized or not
3) rules to determine when and how to make a insert/update to RAX DB

See attachements A thru I at end of this document for the rules for each table.

2.0 Configuration Information

2.1 Apps_defaults tokens

The following apps_defaults tokens are used by this application:

db_name	name of the IHFS database on awips system
pghost	name of the system the IHFS database is on
adb_name	name of the archive database on the archive system
rax_pghost	name of the system the archive database is on
adb_sync_logs_dir	location of the log files (/rfc_arc/logs/dbsync default)
adb_sync_mode	ANALYSIS (default) or UPDATE
adb_sync_tablenames	ALL (default) or a quoted list of table names separated by blanks (e.g. “location adjustfactor crest”). Tables that can be in the list are: location, ingestfilter, reservoir, datalimits, locdatalimits, adjustfactor, rivercrit, crest and rating.
adb_sync_ihfs_ingest	USE (default) or IGNORE
adb_sync_rivercrit	ACTION (default) or FIS or BOTH
adb_sync_ihfs_units	ENGL (default) or METR
adb_sync_ihfs_interpolate	LIN (default) or LOG
adb_sync_ihfs_debug	OFF (default) or ON

To use something other than the above defaults, the user must add the option to their .Apps_defaults_site file that is located in the directory /awips/hydroapps. The following table describes how these tokens are used.

token name	what it controls and how it behaves
adb_sync_logs_dir	The value of this token is used for the directory name for the following files: *tablename*RecordsNew.out *tablename*RecordsDifferent.out RaxSync.log.*yyyymmdd* RaxDbMods.log.*yyyymmdd*
adb_sync_mode	This token controls weather the application is run in Analysis Only or Update mode. Setting this token to Update will run the application in Update mode.
adb_sync_tablenames	This token is used to determine which tables are processed by the application. If set to ALL then all meta-data tables will be processed. If set to a string of table names enclosed in quotes then the application will process only those tables.
adb_sync_ihfs_ingest	This token controls whether or not the ingest column of the IngestFilter table is synchronized. If set to USE, then compare the ingest column. If set to IGNORE, then do not compare the ingest columns and always set the ingest column to 1 when creating a new RAX IngestFilter record.
adb_sync_rivercrit	This token controls whether the IHFS riverStat columns wstg/action_flow are compared to the RAX RiverCrit columns fis/fisf or action/actionf or both sets. If set to FIS, then wstg/action_flow are compared to fis/fisf. If set to ACTION, then wstg/action_flow are compared to action/actionf. If set to BOTH, then wstg/action_flow are compared to both fis/fisf and action/actionf.
adb_sync_ihfs_units	This token controls what value is placed in the units column of the Rating table when inserting a new record in the RAX database.
adb_sync_ihfs_interpolate	This token controls what value is placed in the interpolate column of the Rating table when inserting a new record in the RAX database.
adb_sync_ihfs_debug	This token controls whether the *tablename*Debug.out files are generated or not.
adb_name	This is the name of the RAX database which is used during the synchronization.
db_name	This is the name of the IHFS database which is used during the synchronization.

Table 2. apps_defaults tokens, what they control and how they behave

3.0 User How-To

This application can be run manually or as a cron job.

3.1 Manual runs

To run manually, do the following on the RAX:

cd /rfc_arc/bin [Enter]
run_raxdb_sync [Enter]

The command line prompt returns in the terminal window when the application has finished. All output can be found in the directory specified by the apps_defaults token adb_sync_logs_dir.

3.2 Cron Job

To run as a cron job, add an entry to user oper's cron. The crontab file can be found in directory /rfc_arc/crons and is called adb_oper_crons. Make sure a backup copy of this file is kept, as it is likely this file will be overwritten when new software loads are installed. A cron entry for this application would be similar to the following.

```
15 4 * * * /rfc_arc/bin/run_raxdb_sync process >>/dev/null 2>>/dev/null
```

3.3 First time syncing the rating and ratingshift tables

Prior to build ob8.3 there was no baseline application that allowed the user to easily update the rating and ratingshift db tables. Some RFCs created local applications that allowed them to populate these tables before the tables were redesigned in build ob7.2 and before the change in RDMBS from in Informix to Postgres. Thus, the first time this application is run it will default the rating curve date fields (valid_date for rating table, begin_date for the ratingshift table) to current date and time. The user may need to go in and manually reset these dates to indicate when the rating actually was implemented so that other RAX baseline applications will correctly use the rating data. The applications start_raxbase or adbpg.pl may be used to do this, Sections J-6 and J-3 respectively.

4.0 Log Files

All log files are located in the directory defined by the apps_defaults token adb_sync_logs_dir.

There are two types of files for each table, a "New" and a "Different" file. A "New" file will list all entries found in IHFS DB which do not exist in the RAX DB. A "Different" file will list all entries where, based on the synchronization rules, the application found a difference between the IHFS DB and the RAX DB. Examples of both these types of

files are shown below for the location table.

LocationRecordsNew.out

```
A new Location record will need to be created in RAX db for Primary Key = |ATRS2|2007-10-29|
lid = ATRS2
sbd = 2007-10-29
sed = null
goes = null
name = Buffalo 14 SE
det = Antelope Range
lat = 45.5200
lon = -103.2800
elev = 2890
state = SD
huc = null
countyfips = 063
zon = null
hsa = UNR
post = 2
dbsource = null
rfc = MB
countryfips = US

A new Location record will need to be created in RAX db for Primary Key = |BASK1|2007-09-24|
lid = BASK1
sbd = 2007-09-24
sed = null
goes = null
name = Basehor 3 NE
det = null
lat = 39.1692
lon = -94.9069
elev = 964
state = KS
huc = null
countyfips = 103
zon = null
hsa = EAX
post = 2
dbsource = null
rfc = MB
countryfips = US
```

LocationRecordsDifferent.out

```
Differences found in Table = Location for Primary Key = |0004N8|2000-11-09|
Column = elev   IHFS_value = 2700   Rax_value = 0

Differences found in Table = Location for Primary Key = |0005N8|2000-11-09|
Column = elev   IHFS_value = 2717   Rax_value = 0

Differences found in Table = Location for Primary Key = |0006N8|2000-11-09|
Column = elev   IHFS_value = 2503   Rax_value = 0

Differences found in Table = Location for Primary Key = |0007N8|2000-11-09|
Column = elev   IHFS_value = 2559   Rax_value = 0

Differences found in Table = Location for Primary Key = |0008N8|2000-11-09|
Column = elev   IHFS_value = 2592   Rax_value = 0

Differences found in Table = Location for Primary Key = |0009N8|2000-11-09|
Column = elev   IHFS_value = 2799   Rax_value = 0

Differences found in Table = Location for Primary Key = |0010N8|2000-11-09|
Column = elev   IHFS_value = 2549   Rax_value = 0

Differences found in Table = Location for Primary Key = |0011N8|2000-11-09|
Column = elev   IHFS_value = 2618   Rax_value = 0
```

In addition to the log files for each table, there are two other log files: RaxDbMods.log.”date-stamp” and RaxSync.log.”date-stamp”, where “date-stamp” has the format ccyymmdd (ex. 20071117). A new file is created for each of these files each day that the application is run.

The RaxDbMods.log.”date-stamp” contains a history of all changes that were made to tables (as defined by the token adb_sync_tablenames) when the apps_defaults token adb_sync_mode is set to UPDATE. An example of this output is shown below.

RaxDbMods.log.”date-stamp”

```
2007/11/17 22:38:17 Replaced old  RAX Location record = |0004N8|2000-11-09|null|null|Regent 10
S|Nd Arb Network|46.2667|-102.5500|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
2007/11/17 22:38:17 Updated with  RAX Location record = |0004N8|2000-11-09|2007-11-17|null|Regent
10 S|Nd Arb Network|46.2667|-102.5500|0|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Inserted new  RAX Location record = |0004N8|2007-11-17|null|null|Regent 10
S|Nd Arb Network|46.2667|-102.5500|2700|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Replaced old  RAX Location record = |0005N8|2000-11-09|null|null|Bucyrus 8
N|Nd Arb Network|46.1833|-102.7833|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
2007/11/17 22:38:17 Updated with  RAX Location record = |0005N8|2000-11-09|2007-11-
17|null|Bucyrus 8 N|Nd Arb Network|46.1833|-102.7833|0|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Inserted new  RAX Location record = |0005N8|2007-11-17|null|null|Bucyrus 8
N|Nd Arb Network|46.1833|-102.7833|2717|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Replaced old  RAX Location record = |0006N8|2000-11-09|null|null|Adams 7
NE|Nd Arb Network|46.1667|-102.3833|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
2007/11/17 22:38:17 Updated with  RAX Location record = |0006N8|2000-11-09|2007-11-17|null|Adams
7 NE|Nd Arb Network|46.1667|-102.3833|0|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Inserted new  RAX Location record = |0006N8|2007-11-17|null|null|Adams 7
NE|Nd Arb Network|46.1667|-102.3833|2503|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Replaced old  RAX Location record = |0007N8|2000-11-09|null|null|Adams 14
E|Nd Arb Network|46.1000|-102.2500|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
2007/11/17 22:38:17 Updated with  RAX Location record = |0007N8|2000-11-09|2007-11-17|null|Adams
14 E|Nd Arb Network|46.1000|-102.2500|0|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Inserted new  RAX Location record = |0007N8|2007-11-17|null|null|Adams 14
E|Nd Arb Network|46.1000|-102.2500|2559|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Replaced old  RAX Location record = |0008N8|2000-11-09|null|null|Haynes 15
E|Nd Arb Network|46.0167|-102.1667|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
2007/11/17 22:38:17 Updated with  RAX Location record = |0008N8|2000-11-09|2007-11-17|null|Haynes
15 E|Nd Arb Network|46.0167|-102.1667|0|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Inserted new  RAX Location record = |0008N8|2007-11-17|null|null|Haynes 15
E|Nd Arb Network|46.0167|-102.1667|2592|ND|null|001|null|BIS|BIS|2|null|MB|US|

2007/11/17 22:38:17 Replaced old  RAX Location record = |0009N8|2000-11-09|null|null|Bucyrus 3
NW|Nd Arb Network|46.0833|-102.8167|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
2007/11/17 22:38:17 Updated with  RAX Location record = |0009N8|2000-11-09|2007-11-
17|null|Bucyrus 3 NW|Nd Arb Network|46.0833|-102.8167|0|ND|null|001|null|BIS|BIS|2|null|MB|US|
```

The last log file, RaxSync.log.”date-stamp”, is a general log of the actions the application took that day, no matter what the token adb_sync_mode is set to. An example of this logfile is shown below.

RaxSync.log."date-stamp"

```
################################################################################
2007/11/17 21:27:08 Starting Batch Synchronization program version date 11/14/2007.

IHFS database used for synchronization : hd_ob81krf
RAX  database used for synchronization : adb_ob72krf
Apps_defaults Token adb_sync_mode set to : ANALYSIS
Mode : ANALYSIS ONLY
Apps_defaults Token adb_sync_tablenames set to: locdatalimits
List of Tables to process : LOCDATALIMITS

2007/11/17 21:27:08 Begin Analyzing LocDataLimits Table records...
999 = Number of LocDataLimits records found in the IHFS db
816 = Number of IHFS LocDataLimits records which are 'The Same' as those in the RAX db
168 = Number of IHFS LocDataLimits records which are 'Different' than those in the RAX db
15 = Number of IHFS LocDataLimits records which are NOT found in the RAX db
Listing of Differences found between LocDataLimits tables are found in file
/rfc_arc/logs/dbsync/LocDataLimitsRecordsDifferent.out
Listing of New LocDataLimits records to be added to the RAX db are found in file
/rfc_arc/logs/dbsync/LocDataLimitsRecordsNew.out
2007/11/17 21:27:08 End of Analysis of LocDataLimits Table.

2007/11/17 21:27:08 Batch Synchronization program successfully completed.

################################################################################
2007/11/17 21:32:18 Starting Batch Synchronization program version date 11/14/2007.

IHFS database used for synchronization : hd_ob81krf
RAX  database used for synchronization : adb_ob72krf
Apps_defaults Token adb_sync_mode set to : ANALYSIS
Mode : ANALYSIS ONLY
Apps_defaults Token adb_sync_tablenames set to: datalimits
List of Tables to process : DATALIMITS

2007/11/17 21:32:18 Begin Analyzing DataLimits Table records...
60 = Number of DataLimits records found in the IHFS db
60 = Number of IHFS DataLimits records which are 'The Same' as those in the RAX db
0 = Number of IHFS DataLimits records which are 'Different' than those in the RAX db
0 = Number of IHFS DataLimits records which are NOT found in the RAX db
2007/11/17 21:32:18 End of Analysis of DataLimits Table.

2007/11/17 21:32:18 Batch Synchronization program successfully completed.
```

In addition to the previously mentioned logfiles, there is a debug log file for each table that can be generated if the apps_defaults token adb_sync_ihfs_debug is set to ON. This file generates information that can help the user determine if the rules for a particular table were followed the way the user expected. An example for the location table follows.

LocationDebug.out

```
A new Location record will need to be created in RAX db for Primary Key =  |ATRS2|2007-10-29|
using the following information from the IHFS db:
lid = ATRS2   IHFS_lid = ATRS2
sbd = 2007-10-29   IHFS_sbd = null   IHFS_lrevise = 2007-10-29
sed = null
goes = null
name = Buffalo 14 SE   IHFS_name = BUFFALO   IHFS_detail = 14 SE
det = Antelope Range   IHFS_det = ANTELOPE RANGE
lat = 45.5200   IHFS_lat = 45.5200
lon = -103.2800   IHFS_lon = 103.2800
elev = 2890   IHFS_elev = 2890.0
state = SD   IHFS_state = SD
huc = null
countyfips = 063   IHFS_county = Harding   IHFS_state = SD
zon = null
```

```
hsa = UNR   IHFS_hsa = UNR
post = 2   IHFS_post = 1
dbsource = null
rfc = MB   IHFS_rfc = MBRFC
countryfips = US

A new Location record will need to be created in RAX db for Primary Key = |BASK1|2007-09-24|
using the following information from the IHFS db:
lid = BASK1   IHFS_lid = BASK1
sbd = 2007-09-24   IHFS_sbd = null   IHFS_lrevise = 2007-09-24
sed = null
goes = null
name = Basehor 3 NE   IHFS_name = BASEHOR   IHFS_detail = 3 NE
det = null   IHFS_det = null
lat = 39.1692   IHFS_lat = 39.1692
lon = -94.9069   IHFS_lon = 94.9069
elev = 964   IHFS_elev = 964.0
state = KS   IHFS_state = KS
huc = null
countyfips = 103   IHFS_county = Leavenworth   IHFS_state = KS
zon = null
hsa = EAX   IHFS_hsa = EAX
post = 2   IHFS_post = 1
dbsource = null
rfc = MB   IHFS_rfc = MBRFC
countryfips = US
```

NOTE: The files without date-stamps are overwritten each time the application runs BUT the files with date-stamps are not. The date-stamp files will accumulate over time, and the RFC must either add an entry to the RAX's purge_files program to houseclean them out, archive them elsewhere, or delete them manually to prevent the /rfc_arc partition from filling up.

5.0 Troubleshooting Information

Log files are generated by the applications when run by the run_raxdb_sync script. Check these log files for problems.

If further help is needed, contact the RFC Support Group.

Attachment A - location

Rules for inserting a new database record into the RAX Location table when synchronizing with an existing IHFS Location record for which there is no active* RAX Location record existing with the same lid.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
sbd	date	1) sbd from IHFS if not null 2) lrevise from IHFS if not null 3) Today's date
sed	date	NULL
goes	character varying(8)	NULL
name	character varying(60)	Mixed case version of name + detail from IHFS
det	character varying(40)	Mixed case version of det from IHFS
lat	double precision	lat from IHFS
lon	double precision	lon from IHFS multiplied by -1.0
elev	integer	elev from IHFS + 0.5 then cast as an Int
state	character(2)	state from IHFS unless it is "XX", then try to determine the state from the lid
huc	character varying(8)	NULL
countyfips	character(3)	countynum from IHFS counties table if not null, otherwise set to "XXX"
zon	character(4)	NULL
hsa	character(3)	hsa from IHFS
wfo	character(3)	wfo from IHFS
post	smallint	if post value from IHFS is 1 then set to 2 in RAX, otherwise set to 0
dbsource	character(3)	NULL
rfc	character(2)	first two letters of rfc from IHFS
countryfips	character(2)	Initialize as "US" United States, set to "CA" Canada or "MX" Mexico if it can be determined by the state ID.

* An active (or current) Location record in the RAX database is one that has a NULL value for its sed column.

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the Location table.

RAX Column Name	IHFS Column name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
sbd	sbd	**Not compared** for synchronization purposes
sed		**Not checked**, no corresponding field in IHFS
goes		**Not checked**, no corresponding field in IHFS
name	name, detail	RAX name must be "equal to" IHFS name + detail (note: upper vs. lower case ignored)
det	det	First 30 character in RAX det same as IHFS det (note: upper vs. lower case ignored)
lat	lat	if lat from RAX equals lat from IHFS
lon	lat	if lon from RAX equals lon from IHFS multiplied by -1.0
elev	elev	if RAX value equals IHFS value cast as an Int
state	state	if RAX value equals IHFS value
huc		**Not checked**, no corresponding field in IHFS
countyfips		**Not checked** for synchronization purposes
zon		**Not checked**, no corresponding field in IHFS
hsa	hsa	if RAX value equals IHFS value
wfo	wfo	if RAX value equals IHFS value
post	post	**Not compared** for synchronization purposes
dbsource		**Not checked**, no corresponding field in IHFS
rfc	rfc	rfc from RAX equals first two letters of rfc from IHFS
countryfips		**Not checked**, no corresponding field in IHFS

Rules for determining when and how to automatically insert and update Location records in the RAX database.

A. If there is a Location table record in the IHFS database with a lid for which there is no active***** RAX Location record with the same lid in the RAX database then that record will be inserted into the Location table of the RAX database.

B. If there is an active***** RAX Location record in the RAX database which has a corresponding Location record in the IHFS database and any of the following columns (name, det, lat, lon, elev, state, hsa, wfo or rfc) are different between the two database records then
   1) The sed column in the current RAX Location record will be set to Today's date.
   2) A new RAX Location record will be inserted into the RAX database with the same lid and a sbd set to Today's date. The other columns will be set to the following values:
      - sed set to NULL
      - name, det, lat, lon, elev, state, hsa, wfo and rfc will be set to the values from the IHFS Location record
      - goes, huc, countyfips, zon, post, dbsource, and countryfips will be set to the values from the old RAX Location record

***** An active (or current) Location record in the RAX database is one that has a NULL value for its sed column.

Attachment B - ingestfilter

Rules for inserting a new database record into the RAX IngestFilter table when synchronizing with an exitsting IHFS IngestFilter record for which there is no RAX IngestFilter record existing with the same lid, pe, dur, ts, and extremum.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
pe1	character(1)	First character of pe from IHFS
pe2	character(1)	Second character of pe from IHFS
dur	character(1)	Character value Duration code converted from the numeric value dur from IHFS
idur	smallint	Numeric value dur from IHFS
t	character(1)	First character of ts from IHFS
s	character(1)	Second character of ts from IHFS
e	character(1)	extremum from IHFS
ts_rank	smallint	ts_rank from IHFS
det	character varying(40)	NULL
ingest	smallint	1) If adb_sync_ihfs_ingest token is set to USE then set the RAX ingest column to the number 0(zero) if ingest from IHFS is "F", otherwise set it to 1(one). 2) If adb_sync_ihfs_ingest token is set to IGNORE, then always set the RAX ingest column to the number 1(one) regardless of the value of the IHFS ingest coulmn.
new_report	character(1)	"N"
active	character(1)	"Y"
ofs_input	character(1)	Character "0"(zero) if ofs_input from IHFS is "F", otherwise set it to "1"(one)
obstime	time without time zone	NULL
ownag	character varying(6)	NULL
ownloc	character(3)	NULL
mpe_input	character(1)	Character "0"(zero) if stg2_input from IHFS is "F", otherwise set it to "1"(one)

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the IngestFilter table.

Rax Column Name	IHFS Column name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
pe1	pe (first character)	Part of the key so **they are the same by default**
pe2	pe (second character)	Part of the key so **they are the same by default**
dur		**Not checked**, no corresponding field in IHFS
idur	dur	Part of the key so **they are the same by default**
t	ts (first character)	Part of the key so **they are the same by default**
s	ts (second character)	Part of the key so **they are the same by default**
e	extremum	Part of the key so **they are the same by default**
ts_rank	ts_rank	if RAX value equals IHFS value
det		**Not checked**, no corresponding field in IHFS
ingest	ingest	1) If adb_sync_ihfs_ingest token is set to USE, then check if RAX value is 0 and IHFS value is "F" or if RAX value is 1 and IHFS value is "T" 2) If adb_sync_ihfs_ingest token is set to IGNORE, then the ingest columns are **Not compared** for synchronization purposes.
new_report		**Not checked**, no corresponding field in IHFS
active		**Not checked**, no corresponding field in IHFS
ofs_input	ofs_input	**Not compared** for synchronization purposes
obstime		**Not checked**, no corresponding field in IHFS
ownag		**Not checked**, no corresponding field in IHFS
ownloc		**Not checked**, no corresponding field in IHFS
mpe_input	stg2_input	**Not compared** for synchronization purposes

Rules for determining when and how to automatically insert and update IngestFilter records in the RAX database.

A. If there is an IngestFilter table record in the IHFS database with a lid, pe, dur, ts, and extremum for which there is no corresponding RAX IngestFilter record with the same lid, pe1, pe2, idur, t, s, and e in the RAX database then that record***** will be inserted into the IngestFilter table of the RAX database.

B. If there is a RAX IngestFilter record in the RAX database which has a corresponding IngestFilter record in the IHFS database and any of the following columns (ts_rank or ingest*****) are different between the two database records then
    - The RAX IngestFilter record will be updated with the ts_rank and ingest* column values from the IHFS IngestFilter record.

***** Note: The updating and comparison of the ingest column is controlled by the apps_defaults token adb_sync_ihfs_ingest. These rules are explained in the tables above.

Attachment C - rivercrit

Rules for inserting a new database record into the RAX RiverCrit table when synchronizing with existing IHFS RiverStat & FloodCat records for which there is no RAX RiverCrit record existing with the same lid.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS RiverStat table
pe1 pe2	character(1) character(1)	1) primary_pe from IHFS RiverStat if not NULL 2) Derived from the IngestFilter table if not more than one H* 3) Defaults to HG
vdtime	date	Today's date
lowscreen	double precision	NULL
sigrate	double precision	NULL
screenrate	double precision	NULL
fis	numeric(10.2)	1) wstg from IHFS RiverStat if *adb_sync_rivercrit* token is set to FIS or BOTH 2) NULL if *adb_sync_rivercrit* token is set to ACTION
action	numeric(10.2)	1) wstg from IHFS RiverStat if *adb_sync_rivercrit* token is set to ACTION or BOTH 2) NULL if *adb_sync_rivercrit* token is set to FIS
alert	numeric(10.2)	NULL
bank	numeric(10.2)	bf from IHFS RiverStat
flood	numeric(10.2)	1) fs from IHFS RiverStat if not NULL 2) minor_stage from FloodCat
modflood	numeric(10.2)	moderate_stage from IHFS FloodCat
majflood	numeric(10.2)	major_stage from IHFS FloodCat
record	numeric(10.2)	NULL
highscreen	double precision	NULL
damscreen	double precision	NULL
lowscreenf	double precision	NULL
sigratef	double precision	NULL
screenratef	double precision	NULL
fisf	double precision	1) action_flow from IHFS RiverStat if *adb_sync_rivercrit* token is set to FIS or BOTH 2) NULL if *adb_sync_rivercrit* token is set to ACTION
actionf	double precision	1) action_flow from IHFS RiverStat if *adb_sync_rivercrit* token is set to ACTION or BOTH 2) NULL if *adb_sync_rivercrit* token is set to FIS
alertf	double precision	NULL
bankf	double precision	NULL
floodf	double precision	1) fq from IHFS RiverStat if not NULL

		2) minor_flow from FloodCat
modfloodf	double precision	moderate_flow from IHFS FloodCat
majfloodf	double precision	major_flow from IHFS FloodCat
recordf	double precision	NULL
highscreenf	double precision	NULL
damscreenf	double precision	NULL
sigratet	double precision	NULL
screenratet	double precision	NULL
lowscreenq	character(1)	NULL
sigrateq	character(1)	NULL
screenrateq	character(1)	NULL
fisq	character(1)	NULL
actionq	character(1)	NULL
alertq	character(1)	NULL
bankq	character(1)	NULL
floodq	character(1)	NULL
modfloodq	character(1)	NULL
majfloodq	character(1)	NULL
recordq	character(1)	NULL
highscreenq	character(1)	NULL
damscreenq	character(1)	NULL
stream	character varying(32)	stream from IHFS RiverStat
lat	double precision	lat from IHFS RiverStat
lon	double precision	lon from IHFS RiverStat
da	double precision	da from IHFS RiverStat
mile	double precision	mile from IHFS RiverStat
zd	double precision	zd from IHFS RiverStat
vdatum	character varying(20)	vdatum from IHFS RiverStat
cb	double precision	cb from IHFS RiverStat
level	character varying(20)	level from IHFS RiverStat
pool	double precision	pool from IHFS RiverStat
por	character varying(30)	por from IHFS RiverStat
tide	character varying(8)	tide from IHFS RiverStat
backwater	character varying(8)	backwater from IHFS RiverStat
rrevise	date	rrevise from IHFS RiverStat
rsource	character varying(20)	rsource from IHFS RiverStat
response_time	double precision	response_time from IHFS RiverStat
threshold_runoff	double precision	threshold_runoff from IHFS RiverStat
uhgdur	integer	uhgdur from IHFS RiverStat
remark	character varying(255)	remark from IHFS RiverStat

Rules for determining if a database record is synchronized ("the same") between the RAX RiverCrit table and the IHFS RiverStat & FloodCat tables.

Rax Column Name	IHFS Column name	What makes them "in synch"
lid	lid in RiverStat	Part of the key so **they are the same by default**
pe1 pe2	primary_pe (first character) primary_pe (second character)	This comparison is only done if the primary_pe of the RiverStat table is NOT NULL.
vdtime		**Not checked**, no corresponding field in IHFS
lowscreen		**Not checked**, no corresponding field in IHFS
sigrate		**Not checked**, no corresponding field in IHFS
screenrate		**Not checked**, no corresponding field in IHFS
fis	wstg	These two columns are compared only if the *adb_sync_rivercrit* token is set to FIS or BOTH
action	wstg	These two columns are compared only if the *adb_sync_rivercrit* token is set to ACTION or BOTH
alert		**Not checked**, no corresponding field in IHFS
bank	bf	if RAX value equals IHFS value
flood	fs (RiverStat) OR minor_stage (FloodCat)	RiverCrit flood is compared to: 1) fs from IHFS RiverStat if not NULL 2) minor_stage from FloodCat
modflood	moderate_stage	if RAX value equals IHFS value
majflood	major_stage	if RAX value equals IHFS value
record		**Not checked**, no corresponding field in IHFS
highscreen		**Not checked**, no corresponding field in IHFS
damscreen		**Not checked**, no corresponding field in IHFS
lowscreenf		**Not checked**, no corresponding field in IHFS
sigratef		**Not checked**, no corresponding field in IHFS
screenratef		**Not checked**, no corresponding field in IHFS
fisf	action_flow	These two columns are compared only if the *adb_sync_rivercrit* token is set to FIS or BOTH
actionf	action_flow	These two columns are compared only if the *adb_sync_rivercrit* token is set to ACTION or BOTH
alertf		**Not checked**, no corresponding field in IHFS
bankf		**Not checked**, no corresponding field in IHFS
floodf	fq (RiverStat) OR minor_flow (FloodCat)	RiverCrit floodf is compared to: 1) fq from IHFS RiverStat if not NULL 2) minor_flow from FloodCat
modfloodf	moderate_flow	if RAX value equals IHFS value
majfloodf	major_flow	if RAX value equals IHFS value
recordf		**Not checked**, no corresponding field in IHFS
highscreenf		**Not checked**, no corresponding field in IHFS
damscreenf		**Not checked**, no corresponding field in IHFS
sigratet		**Not checked**, no corresponding field in IHFS
screenratet		**Not checked**, no corresponding field in IHFS
lowscreenq		**Not checked**, no corresponding field in IHFS

sigrateq		**Not checked**, no corresponding field in IHFS
screenrateq		**Not checked**, no corresponding field in IHFS
fisq		**Not checked**, no corresponding field in IHFS
actionq		**Not checked**, no corresponding field in IHFS
alertq		**Not checked**, no corresponding field in IHFS
bankq		**Not checked**, no corresponding field in IHFS
floodq		**Not checked**, no corresponding field in IHFS
modfloodq		**Not checked**, no corresponding field in IHFS
majfloodq		**Not checked**, no corresponding field in IHFS
recordq		**Not checked**, no corresponding field in IHFS
highscreenq		**Not checked**, no corresponding field in IHFS
damscreenq		**Not checked**, no corresponding field in IHFS
stream	stream	if RAX value equals IHFS value
lat	lat	if RAX value equals IHFS value
lon	lon	if RAX value equals IHFS value
da	da	if RAX value equals IHFS value
mile	mile	if RAX value equals IHFS value
zd	zd	if RAX value equals IHFS value
vdatum	vdatum	if RAX value equals IHFS value
cb	cb	if RAX value equals IHFS value
level	level	if RAX value equals IHFS value
pool	pool	if RAX value equals IHFS value
por	por	if RAX value equals IHFS value
tide	tide	if RAX value equals IHFS value
backwater	backwater	if RAX value equals IHFS value
rrevise	rrevise	if RAX value equals IHFS value
rsource	resource	if RAX value equals IHFS value
response_time	response_time	if RAX value equals IHFS value
threshold_runoff	threshold_runoff	if RAX value equals IHFS value
uhgdur	uhgdur	if RAX value equals IHFS value
remark	remark	if RAX value equals IHFS value

NOTE: The IHFS RiverStat & FloodCat records are compared to the RAX RiverCrit record which has the latest vdtime for the same lid.

Rules for determining when and how to automatically insert and update RiverCrit records in the RAX database.

A. If there is a RiverStat table record in the IHFS database with a lid for which there is no corresponding RAX RiverCrit record with the same lid in the RAX database then that record will be inserted into the RiverCrit table of the RAX database with a vdtime set to Today's date.

B. If there is a RiverCrit record in the RAX database which has a corresponding RiverStat record in the IHFS database and any of the following columns (fis, action, bank, flood, modflood, majflood, fisf, actionf, floodf, modfloodf, majfloodf, stream, lat, lon, da, mile, zd, vdatum, cb, level, pool, por, tide, backwater, rrevise, rsource, response_time, threshold_runoff, uhgdur, or remark) are different between the two database records then

1) A new RAX RiverCrit record will be inserted into the RAX database with the same lid and a vdtime set to Today's date. The other columns will be set to the following values:

- pe1, pe2, bank, stream, lat, lon, da, mile, zd, vdatum, cb, level, pool, por, tide, backwater, rrevise, rsource, response_time, threshold_runoff, uhgdur, and remark will be set to the values from the IHFS RiverStat record
- modflood, majflood, modfloodf, majfloodf will be set to the values from the IHFS FloodCat record
- flood and floodf will be set to the values from either the IHFS RiverStat or Floodcat record depending if there is a NULL value found. (See table above)
- fis, action, fisf and actionf will be set to values from the IHFS RiverStat record depending on the value of the apps-defaults token *adb_sync_rivercrit*. (See table above)
- lowscreen, sigrate, screenrate, alert, record, highscreen, damscreen, lowscreenf, sigratef, screenratef, alertf, bankf, recordf, highscreenf, damscreenf, sigratet, screenratet, lowscreenq, sigrateq, screenrateq, fisq, actionq, alertq, bankq, floodq, modfloodq, majfloodq, recordq, highscreenq, and damscreenq will be set to the values from the old RAX RiverCrit record

Attachment D - crest

Rules for inserting a new database record into the RAX Crest table when synchronizing with an existing IHFS Crest record for which there is no RAX Crest record existing with the same lid and datcrst.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
datecrst	date	datcrst from IHFS
crstdatetime	character varying(5)	timcrst from IHFS
stage	double precision	stage from IHFS
stg_qual	character(1)	NULL
flow	double precision	q from IHFS cast as a double
flow_qual	character(1)	NULL
hw	character(1)	hw from IHFS
jam	character(1)	jam from IHFS
olddatum	character(1)	olddatum from IHFS
prelim	character(1)	prelim from IHFS

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the Crest table.

RAX Column Name	IHFS Column name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
datecrst	datcrst	Part of the key so **they are the same by default**
crstdatetime	timcrst	if RAX value equals IHFS value
stage	stage	if RAX value equals IHFS value
stg_qual		**Not checked**, no corresponding field in IHFS
flow	q	if RAX value equals IHFS value cast as a double
flow_qual		**Not checked**, no corresponding field in IHFS
hw	hw	if RAX value equals IHFS value
jam	jam	if RAX value equals IHFS value
olddatum	olddatum	if RAX value equals IHFS value
prelim	prelim	if RAX value equals IHFS value

Rules for determining when and how to automatically insert and update Crest records in the RAX database.

A. If there is a Crest table record in the IHFS database with a lid and datcrst for which there is no corresponding RAX Crest record with the same lid and datecrst in the RAX database then that record will be inserted into the Crest table of the RAX database.

B. If there is a RAX Crest record in the RAX database which has a corresponding Crest record in the IHFS database and any of the following columns (crstdatetime, stage, flow, hw, jam, olddatum or prelim) are different between the two database records then

- The RAX Crest record will be updated with the timcrst, stage, q, hw, jam, olddatum and prelim column values from the IHFS Crest record.

Attachment E - datalimits

Last Edited: October 14, 2007

Rules for inserting a new database record into the RAX DataLimits table when synchronizing with an existing IHFS DataLimits record for which there is no RAX DataLimits record existing with the same pe, dur and monthdaystart.

Column Name	Data Type	Value to use when inserting NEW database record
pe1	character(1)	First character of pe from IHFS
pe2	character(1)	Second character of pe from IHFS
dur	character(1)	Character value Duration code converted from the numeric value dur from IHFS
idur	smallint	Numeric value dur from IHFS
monthdaystart	character varying(5)	monthdaystart from IHFS
monthdayend	character varying(5)	monthdayend from IHFS
gross_range_min	double precision	gross_range_min from IHFS
gross_range_max	double precision	gross_range_max from IHFS
reason_range_min	double precision	reason_range_min from IHFS
reason_range_max	double precision	reason_range_max from IHFS
roc_max	double precision	roc_max from IHFS
alert_limit	double precision	alert_upper_limit from IHFS *****
alert_roc_limit	double precision	alert_roc_limit from IHFS
alarm_limit	double precision	alarm_upper_limit from IHFS *****
alarm_roc_limit	double precision	alarm_roc_limit from IHFS

*** Assumption is that you are interested in the upper limit to check if a value or rate is exceeded.**

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the DataLimits table.

RAX Column Name	IHFS Column name	What makes them "in synch"
pe1	pe (first character)	Part of the key so **they are the same by default**
pe2	pe (second character)	Part of the key so **they are the same by default**
dur	dur (character value)	**Not compared** for synchronization purposes
idur	dur	Part of the key so **they are the same by default**
monthdaystart	monthdaystart	Part of the key so **they are the same by default**
monthdayend	monthdayend	if RAX value equals IHFS value
gross_range_min	gross_range_min	if RAX value equals IHFS value
gross_range_max	gross_range_max	if RAX value equals IHFS value
reason_range_min	reason_range_min	if RAX value equals IHFS value
reason_range_max	reason_range_max	if RAX value equals IHFS value
roc_max	roc_max	if RAX value equals IHFS value
alert_limit	alert_upper_limit *	if RAX alert_limit equals IHFS alert_upper_limit *
alert_roc_limit	alert_roc_limit	if RAX value equals IHFS value
alarm_limit	alarm_upper_limit *	if RAX alarm_limit equals IHFS alarm_upper_limit *
alarm_roc_limit	alarm_roc_limit	if RAX value equals IHFS value

*** Assumption is that you are interested in the upper limit to check if a value or rate is exceeded.**

Rules for determining when and how to automatically insert and update DataLimits records in the RAX database.

A. If there is a DataLimits table record in the IHFS database with a pe, dur and monthdaystart for which there is no corresponding RAX DataLimits record with the same pe1 + pe2, idur and monthdaystart in the RAX database then that record will be inserted into the DataLimits table of the RAX database.

B. If there is a RAX DataLimits record in the RAX database which has a corresponding DataLimits record in the IHFS database and any of the following columns (monthdayend, gross_range_min, gross_range_max, reason_range_min, reason_range_max, roc_max, alert_limit, alert_roc_limit, alarm_limit, or alarm_roc_limit) are different between the two database records then

- The RAX DataLimits record will be updated with the monthdayend, gross_range_min, gross_range_max, reason_range_min, reason_range_max, roc_max, alert_limit, alert_roc_limit, alarm_limit, and alarm_roc_limit column values from the IHFS DataLimits record.

Attachment F - locdatalimits

Last Edited: October 15, 2007

Rules for inserting a new database record into the RAX LocDataLimits table when synchronizing with an existing IHFS LocDataLimits record for which there is no RAX LocDatalimits record existing with the same lid, pe, dur and monthdaystart.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
pe1	character(1)	First character of pe from IHFS
pe2	character(1)	Second character of pe from IHFS
dur	character(1)	Character value Duration code converted from the numeric value dur from IHFS
idur	smallint	Numeric value dur from IHFS
monthdaystart	character varying(5)	monthdaystart from IHFS
monthdayend	character varying(5)	monthdayend from IHFS
gross_range_min	double precision	gross_range_min from IHFS
gross_range_max	double precision	gross_range_max from IHFS
reason_range_min	double precision	reason_range_min from IHFS
reason_range_max	double precision	reason_range_max from IHFS
roc_max	double precision	roc_max from IHFS
alert_limit	double precision	alert_upper_limit from IHFS *****
alert_roc_limit	double precision	alert_roc_limit from IHFS
alarm_limit	double precision	alarm_upper_limit from IHFS *****
alarm_roc_limit	double precision	alarm_roc_limit from IHFS

*** Assumption is that you are interested in the upper limit to check if a value or rate is exceeded.**

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the LocDataLimits table.

Rax Column Name	IHFS Column name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
pe1	pe (first character)	Part of the key so **they are the same by default**
pe2	pe (second character)	Part of the key so **they are the same by default**
dur	dur (character value)	**Not compared** for synchronization purposes
idur	dur	Part of the key so **they are the same by default**
monthdaystart	monthdaystart	Part of the key so **they are the same by default**
monthdayend	monthdayend	if RAX value equals IHFS value
gross_range_min	gross_range_min	if RAX value equals IHFS value
gross_range_max	gross_range_max	if RAX value equals IHFS value
reason_range_min	reason_range_min	if RAX value equals IHFS value
reason_range_max	reason_range_max	if RAX value equals IHFS value
roc_max	roc_max	if RAX value equals IHFS value
alert_limit	alert_upper_limit *	if RAX alert_limit equals IHFS alert_upper_limit *
alert_roc_limit	alert_roc_limit	if RAX value equals IHFS value
alarm_limit	alarm_upper_limit *	if RAX alarm_limit equals IHFS alarm_upper_limit *
alarm_roc_limit	alarm_roc_limit	if RAX value equals IHFS value

*** Assumption is that you are interested in the upper limit to check if a value or rate is exceeded.**

Rules for determining when and how to automatically insert and update LocDataLimits records in the RAX database.

A. If there is a LocDataLimits table record in the IHFS database with a lid, pe, dur and monthdaystart for which there is no corresponding RAX LocDataLimits record with the same lid, pe1 + pe2, idur and monthdaystart in the RAX database then that record will be inserted into the LocDataLimits table of the RAX database.

B. If there is a RAX LocDataLimits record in the RAX database which has a corresponding LocDataLimits record in the IHFS database and any of the following columns (monthdayend, gross_range_min, gross_range_max, reason_range_min, reason_range_max, roc_max, alert_limit, alert_roc_limit, alarm_limit, or alarm_roc_limit) are different between the two database records then

- The RAX LocDataLimits record will be updated with the monthdayend, gross_range_min, gross_range_max, reason_range_min, reason_range_max, roc_max, alert_limit, alert_roc_limit, alarm_limit, and alarm_roc_limit column values from the IHFS LocDataLimits record.

Attachment G - adjustfactor

Rules for inserting a new database record into the RAX AdjustFactor table when synchronizing with an existing IHFS AdjustFactor record for which there is no RAX AdjustFactor record existing with the same lid, pe, dur, ts and extremum.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
pe1	character(1)	First character of pe from IHFS
pe2	character(1)	Second character of pe from IHFS
dur	character(1)	Character value Duration code converted from the numeric value dur from IHFS
idur	smallint	Numeric value dur from IHFS
t	character(1)	First character of ts from IHFS
s	character(1)	Second character of ts from IHFS
e	character(1)	extremum from IHFS
begin_date	date	Today's date
divisor	double precision	divisor from IHFS
base	double precision	base from IHFS
multiplier	double precision	multiplier from IHFS
adder	double precision	adder from IHFS

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the AdjustFactor table.

Rax Column Name	IHFS Column name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
pe1	pe (first character)	Part of the key so **they are the same by default**
pe2	pe (second character)	Part of the key so **they are the same by default**
dur	dur (character value)	**Not compared** for synchronization purposes
idur	dur	Part of the key so **they are the same by default**
t	ts (first character)	Part of the key so **they are the same by default**
s	ts (second character)	Part of the key so **they are the same by default**
e	extremum	Part of the key so **they are the same by default**
begin_date		**Not checked,** no corresponding field in IHFS
divisor	divisor	if RAX value equals IHFS value
base	base	if RAX value equals IHFS value
multiplier	multiplier	if RAX value equals IHFS value
adder	adder	if RAX value equals IHFS value

NOTE: The IHFS AdjustFactor record is compared to the RAX AdjustFactor record which has the latest begin_date for the same lid, pe, dur, ts and extremum.

Rules for determining when and how to automatically insert and update AdjustFactor records in the RAX database.

A. If there is an AdjustFactor table record in the IHFS database with a lid, pe, dur, ts and extremum for which there is no corresponding RAX AdjustFactor record with the same lid, pe1 + pe2, idur, t, s and e in the RAX database then that record will be inserted into the AdjustFactor table of the RAX database with a begin_date set to Today's date.

B. If there is a RAX AdjustFactor record in the RAX database which has a corresponding AdjustFactor record in the IHFS database and any of the following columns (divisor, base, multiplier or adder) are different between the two database records then

- A new RAX AdjustFactor record will be added with a begin_date set to Today's date and the rest of the columns set to the values from the IHFS AdjustFactor record.

Attachment H – reservoir

Rules for inserting a new database record into the RAX Reservoir table when synchronizing with an existing IHFS Reservoir record for which there is no active***** RAX Reservoir record existing with the same lid.

Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
sbd	date	Today's date
sed	date	NULL
name	character(20)	Mixed case version of name from IHFS
type	character varying(10)	type from IHFS
owner	character varying(10)	owner from IHFS
deadpool	double precision	deadpool from IHFS
conserpool	double precision	conserpool from IHFS
floodpool	double precision	floodpool from IHFS
spillway	double precision	spillway from IHFS
sill	double precision	sill from IHFS
top	double precision	top from IHFS
surchg	double precision	surchg from IHFS
elev	double precision	elev from IHFS
gates	integer	gates from IHFS
impounded	date	impounded from IHFS
uses	character varying(8)	uses from IHFS

***** An active (or current) Reservoir record in the RAX database is one that has a NULL value for its sed column.

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the Reservoir table.

RAX Column Name	IHFS Column name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
sbd		**Not checked**, no corresponding field in IHFS
sed		**Not checked**, no corresponding field in IHFS
name	name	RAX name must be "equal to" IHFS name (note: upper vs. lower case ignored)
type	type	if RAX value equals IHFS value
owner	owner	if RAX value equals IHFS value
deadpool	deadpool	if RAX value equals IHFS value
conserpool	conserpool	if RAX value equals IHFS value
floodpool	floodpool	if RAX value equals IHFS value
spillway	spillway	if RAX value equals IHFS value
sill	sill	if RAX value equals IHFS value
top	top	if RAX value equals IHFS value
surchg	surchg	if RAX value equals IHFS value
elev	elev	if RAX value equals IHFS value
gates	gates	if RAX value equals IHFS value
impounded	impounded	if RAX value equals IHFS value
uses	uses	if RAX value equals IHFS value

Rules for determining when and how to automatically insert and update Reservoir records in the RAX database.

A. If there is a Reservoir table record in the IHFS database with a lid for which there is no active***** RAX Reservoir record with the same lid in the RAX database then that record will be inserted into the Reservoir table of the RAX database.

B. If there is an active***** RAX Reservoir record in the RAX database which has a corresponding Reservoir record in the IHFS database and any of the following columns (name, type, owner, deadpool, conserpool, floodpool, spillway, sill, top, surchg, elev, gates, impounded or uses) are different between the two database records then
   1. The sed column in the current RAX Reservoir record will be set to Today's date.
   2. A new RAX Reservoir record will be inserted into the RAX database with the same lid and a sbd set to Today's date. The other columns will be set to the following values:
      - sed set to NULL
      - name, type, owner, deadpool, conserpool, floodpool, spillway, sill, top, surchg, elev, gates, impounded and uses columns will be set to the values from the IHFS Reservoir record.

***** An active (or current) Reservoir record in the RAX database is one that has a NULL value for its sed column.

Attachment I – rating and ratingshift

Rules for inserting a new database record into both the RAX Rating and Rating Shift tables when synchronizing with an existing IHFS Rating record for which there is no RAX Rating record existing with the same lid and a src column equal to "IHFS".

RAX Rating Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	lid from IHFS
pe1	character(1)	H
pe2	character(1)	G
tbl	numeric(6,2)	1.0
valid_date	timestamp without timezone	Current Date & Time
src	character varying(10)	"IHFS"
othagid	character varying(16)	gsno from IHFS RiverStat
rfs_input	character(1)	ofs_input from IHFS IngestFilter. Y if ofs_input is T (true) otherwise use N.
stgflow	double precision[]	stages & discharges from IHFS
units	character(4)	Controled by the *adb_sync_ihfs_units* token. ENGL (default) or METR.
interpolate	character(3)	Controled by the *adb_sync_ihfs_interpolate* token. LIN (default) or LOG.
offsets	double precision[]	NULL
allowstg	double precision	NULL

RAX Rating Shift Column Name	Data Type	Value to use when inserting NEW database record
lid	character varying(8)	Same value as the Rating record above
pe1	character(1)	Same value as the Rating record above
pe2	character(1)	Same value as the Rating record above
tbl_ver	numeric(6,2)	Same value as the Rating record above
begin_date	timestamp without timezone	Same value as the Rating record above
src	character varying(10)	Same value as the Rating record above
val_a	double precision	-9999.0 or 0.0 if there is no IHFS Rating Shift
sh_a	double precision	shift_amount from IHFS Rating or 0.0 if null
val_b	double precision	0.0
sh_b	double precision	0.0
val_c	double precision	0.0
sh_c	double precision	0.0
val_d	double precision	0.0
sh_d	double precision	0.0
datum_adj	double precision	0.0

Rules for determining if a database record is synchronized ("the same") between the RAX and IHFS databases for the Rating table.

Rax Rating Column Name	IHFS Column Name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
pe1		**Not checked**, no corresponding field in IHFS
pe2		**Not checked**, no corresponding field in IHFS
tbl		**Not checked**, no corresponding field in IHFS
valid_date		**Not checked**, no corresponding field in IHFS
src		**Not checked**, no corresponding field in IHFS
othagid	othagid	if RAX othagid equals gsno from IHFS RiverStat
rfs_input	ofs_input	**Not compared** for synchronization purposes
stgflow	stage & discharge	if ALL of the stage/discharge pairs from IHFS are found in the RAX
units		**Not checked**, no corresponding field in IHFS
interpolate		**Not checked**, no corresponding field in IHFS
offsets		**Not checked**, no corresponding field in IHFS
allowstg		**Not checked**, no corresponding field in IHFS

NOTE: The IHFS Rating records are compared to the RAX Rating record with the latest valid_date for the same lid and a src column equal to "IHFS".

RAX Rating Shift Column Name	IHFS Column Name	What makes them "in synch"
lid	lid	Part of the key so **they are the same by default**
pe1		**Not checked**, no corresponding field in IHFS
pe2		**Not checked**, no corresponding field in IHFS
tbl_ver		**Not checked**, no corresponding field in IHFS
begin_date	date	**Not compared** for synchronization purposes
src		**Not checked**, no corresponding field in IHFS
val_a		**Not checked**, no corresponding field in IHFS
sh_a	shift_amount	if RAX sh_a equals IHFS shift_amount
val_b		**Not checked**, no corresponding field in IHFS
sh_b		**Not checked**, no corresponding field in IHFS
val_c		**Not checked**, no corresponding field in IHFS
sh_c		**Not checked**, no corresponding field in IHFS
val_d		**Not checked**, no corresponding field in IHFS
sh_d		**Not checked**, no corresponding field in IHFS
datum_adj		**Not checked**, no corresponding field in IHFS

NOTE: The RAX Rating Shift record is only checked if its corresponding RAX Rating record is different than the IHFS Rating record (see above).

Rules for determining when and how to automatically insert and update Rating and Rating Shift records in the RAX database.

A. If there is a group of Rating table (stage & discharge) records in the IHFS database with a lid for which there is no corresponding RAX Rating record with the same lid and a src column equal to "IHFS" in the RAX database then that group of (stage & discharge) records will be used to create a single Rating record in the RAX database along with the default values for the other columns as described in the table above. Also, a new Rating Shift record will be created in the RAX database with the same Primary Key as the Rating record using the rating shift value from the IHFS database if one exists for this lid.

B. If there is a RAX Rating record in the RAX database which has a corresponding Rating record in the IHFS database and any of the discharge values are different or missing for a corresponding IHFS stage value then

C. A new RAX Rating record will be inserted into the RAX database with the same lid and a valid_date set to the current date/time. The other columns will be set to the following values:
   1. stgflow and othagid will be set to values from the IHFS Rating record
   2. pe1, pe2, tbl, src, othagid, rfs_input, units, interpolate, offsets, and allowstg will be set to the values from the old RAX Rating record

D. A new RAX Rating Shift record will be inserted into the RAX database with the same Primary Key as the new RAX Rating record above. The other columns will be set to the following values:
   1. val_a and sh_a will be set to -9999.0 and the shift_amount value from the IHFS Rating Shift record if it exists or else use the values from the old RAX Rating Shift record.
   2. val_b, sh_b, val_c, sh_c, val_d, sh_d, and datum_adj will be set to the values from the old RAX Rating Shift record