
**COSPAS-SARSAT
INTERNATIONAL 406 MHZ
BEACON REGISTRATION DATABASE
(IBRD)**

System Maintenance Manual

C/S D.003
Issue 1 - Revision 1
October 2013



**INTERNATIONAL 406 MHz BEACON
REGISTRATION DATABASE (IBRD)**

System Maintenance Manual

History

| <u>Issue</u> | <u>Revision</u> | <u>Date</u> | Revised Pages | Comments |
|--------------|-----------------|---------------|---------------|-----------------|
| 1 | | November 2005 | All new | Approved CSC-35 |
| 1 | 1 | October 2013 | All new | Approved CSC-51 |

TABLE OF CONTENTS

| | Page |
|---|-------------|
| History | i |
| Table of Contents | ii |
| List of Annexes | iii |
| | |
| 1. INTRODUCTION..... | 1-1 |
| 1.1 Purpose | 1-1 |
| 1.2 Background | 1-1 |
| 1.3 Document Organization | 1-2 |
| | |
| 2. RUNTIME ENVIRONMENT | 2-1 |
| 2.1 Main Application – JRun IBRD Server | 2-1 |
| 2.2 Apache Server | 2-1 |
| 2.3 SQL Server | 2-1 |
| 2.4 Request for Confirmation Process..... | 2-2 |
| 2.5 Purging of Temporary Files | 2-2 |
| 2.6 Database Archival | 2-3 |
| 2.7 System Downtime and “User Warning” Function | 2-4 |
| 2.8 Installing Software Updates | 2-5 |
| | |
| 3. SYSTEM CONFIGURATION | 3-1 |
| 3.1 SystemCfg Table | 3-1 |
| 3.2 MidInfoCfg Table | 3-2 |
| 3.3 PocInfoCfg Table | 3-3 |
| 3.4 Email Support..... | 3-4 |
| | |
| 4. LOGGING | 4-1 |
| 4.1 ConfigChangeLog Table | 4-1 |
| 4.2 OperMsgLog Table | 4-2 |
| 4.3 LogUserAccess Table | 4-4 |
| 4.4 LogQueryAccess Table | 4-4 |
| 4.5 LogPrtEmailFax Table | 4-5 |
| 4.6 Text Based Log Files..... | 4-6 |
| | |
| 5. OTHER SUPPORTING TABLES | 5-1 |
| 5.1 Users Table..... | 5-1 |
| 5.2 Feedback Table | 5-2 |
| 5.3 Reporting Tables | 5-3 |

LIST OF ANNEXES

ANNEX A: LIST OF ABBREVIATIONS AND ACRONYMS A-1

page left blank

1. INTRODUCTION

1.1 Purpose

This document provides information to support maintenance operations for the International 406 MHz Beacon Registration Database (IBRD). The IBRD is a web based application that provides a full featured system for storing and querying data pertaining to the registration of 406 MHz distress beacons.

1.2 Background

Cospas-Sarsat Participants operate a satellite system capable of detecting and locating distress alert transmissions from radio beacons operating at 121.5, 243 and 406 MHz. The beacon signals transmitted over 121.5 MHz and 243 MHz do not include any identification that can be processed by the receiving stations of the Cospas-Sarsat system. Therefore, there is no operational advantage to registering these types of beacons.

The Cospas-Sarsat 406 MHz system provides search and rescue (SAR) services with distress alerts that include the unique 15-character hexadecimal identification of the transmitting beacon. This beacon identification can be decoded to obtain information including:

- a) the type of beacon, i.e. aircraft Emergency Locator Transmitter (ELT), vessel Emergency Position Indicating Radio Beacon (EPIRB) or Personal Locator Beacon (PLB),
- b) the country code and identification data which form the unique beacon identification, and
- c) the type of auxiliary radio locating (homing) device.

If a beacon is properly registered, the 15-character hexadecimal identification of the beacon can be used to access additional information. Beacon registration databases can provide information of great use to SAR services, including:

- a) specific aircraft or vessel identification information,
- b) the make/model of aircraft or vessel in distress,
- c) communications equipment available, and
- d) the number of persons onboard.

Such information can be made available to SAR services only if the required information is provided to the registration authority by the beacon owner/operator.

Registration of 406 MHz beacons is required in accordance with international regulations on SAR established by the International Civil Aviation Organization (ICAO) and the International Maritime Organization (IMO), and registration information must be made available to SAR services on a 24-hour basis. A number of countries have made 406 MHz beacon registration mandatory and maintain national 406 MHz beacon registration databases.

However, despite the clear advantages of registration, a large number of 406 MHz beacons are not properly registered due to a lack of registration facilities in a number of countries.

Furthermore, a number of beacon registries do not have 24-hour points-of-contact easily accessible by SAR services. The IBRD is freely available to users with no access to national registration facilities and to Administrations who wish to avail themselves of the facility to make their national beacon registration data more available to SAR services.

The IBRD provides various levels of access to:

- a) beacon owners who wish to register their beacons,
- b) Administrations who wish to make registration data available to international SAR services, and
- c) SAR services that need to access beacon registration data to efficiently process distress alerts.

Cospas-Sarsat provides the IBRD solely for the purpose of assisting SAR Services in SAR operations and is not intended to fulfill the obligation of National Administrations, as required by IMO and ICAO, to provide a National beacon registration facility.

1.3 Document Organization

Section 2 provides information regarding the Runtime Environment. Descriptions and usage of the major components that comprise the IBRD System are discussed.

Section 3 discusses various aspects pertaining to IBRD System Configuration.

Section 4 provides a background on the various Logging functions.

Section 5 describes Other Tables and supporting functions that may be of interest at the operational level.

- END OF SECTION 1 -

2. RUNTIME ENVIRONMENT

2.1 Main Application – JRun IBRD Server

Macromedia's JRun4 provides the shell that runs the core application software of the IBRD System which in turn provides the actual IBRD Web Site. The source code is "compiled" into formats called Java Archives which are loaded and executed on demand by the internal Java Virtual Machine (JVM) within the JRun4 Server.

The IBRD application, as well as this JRun4 shell, is actually executed by starting a Windows "Service", which is named "JRun IBRD Server". This Windows Service is configured to start up automatically when the Application Server computer itself is powered up. If the service needs to be started or stopped manually (e.g. to install an update) the following steps apply:

1. Log on to the Application Server.
2. On the Start Menu, click Settings>Control Panel>Administrative Tools, and then select Services.
3. Select the "JRun IBRD Server" from the list that appears.
4. To start the server, right-click the server name and then click Start. Once started, the status will change to be Started.
5. To stop the server, right-click the server name and then click Stop.

2.2 Apache Server

The Apache Server is a Hyper Text Transfer Protocol (HTTP) Server product that processes HTTP requests and responses from and to the Internet. In effect, it provides the software level link that connects users on the Internet to the application software running under the JRun Server discussed just above.

The Apache Server is the only IBRD System software that resides on the Web Server computer. It is configured to start up automatically when the Web Server computer itself is powered up. If Apache needs to be started or stopped manually (e.g. to install an update) the following steps apply:

1. Log on to the Web Server.
2. On the Start menu, click Programs>Apache>Start (or Stop, Restart as appropriate).

2.3 SQL Server

Microsoft SQL Server provides the platform that supports the IBRD database. The main application software, running as described above under Section 2.1, uses Java DataBase Connectivity (JDBC) to interface with this database.

SQL Server is configured to start up automatically when the Application Server computer itself is powered up. Independently starting and stopping this service is considered to be an advanced operation beyond the scope of this documentation. Microsoft SQL Server 2000

documentation and references are available through the Microsoft website and available through SQL Server's Books Online installed on the servers.

2.4 Request for Confirmation Process

The Request for Confirmation Process runs in the background in the form of an operating system batch program named "ProcessTwoYearRequest.bat". The purpose of this process is to generate emails to beacon owners whose records have not been updated in about two or more years. The email message specifically requests these beacon owners to access the IBRD System and confirm or accordingly update the information in the database. More details regarding the underlying logic and algorithms associated with this process can be found in the IBRD Software Maintenance Manual.

This batch program is set up to run as a Windows Scheduled Task. To change the scheduled time that the batch program runs, follow the steps below:

Log on to the Application Server.

On the Start menu, click Settings>Control Panel, and then double-click Scheduled Tasks.

Right-click on the batch program task icon and then click Properties.

Make any necessary changes to the task time and day settings and save your changes.

2.5 Purging of Temporary Files

2.5.1 IBRDFileArcPurge Application

The file archive and purge application (IBRDFileArcPurge.exe) is a simple program that identifies various temporary IBRD system files (e.g. text based log files) and removes old (or aged-out) files based on the date. (Although the ability to archive files is available the configuration for the IBRD generally simply purges files). The folders (or file paths) and the number of days to retain a given type of file is configured using the table named "DbmnFileArcPurgeCfg" which resides in the IBRD database. The minimal details on how to configure this process are given below with further details regarding the underlying logic and algorithms provided in the IBRD Software Maintenance Manual.

This application is set up to run as a Windows Scheduled Task. To change the scheduled time that the batch program runs, apply similar steps as given in Section 2.4 above.

2.5.2 DbmnFileArcPurgeCfg Table

2.5.2.1 Purpose

The DbmnFileArcPurgeCfg table provides configuration information to the file archive and purge application. The basic schema of the table is given in the following section and a section below provides example settings and further information.

2.5.2.2 Table Layout

| Field Name | Type | Bytes | Description |
|-------------|---------------------|-------|---|
| Source | varchar | 64 | Folder to check for expired items. |
| InUse | bit | 1 | True indicates that this entry (or row in this table) is active. False indicates that the entry will be ignored by the archive / purge process. |
| Action | varchar | 32 | AArchive@ to move expired items. APurge@ to deleted expired items. |
| Threshold | int | 4 | Age in days before items expire. |
| ItemType | varchar | 32 | AFiles@ to check for expired files. AFolders@ to check for expired folders. |
| Pattern | varchar nullable | 32 | Pattern to match for file names. Value ignored when item=@Folder@. |
| Destination | varchar nullable | 64 | Folder to send expired item to. Value ignored when action=@Purge@. |

Primary Key – Source

2.5.2.3 Example Settings

Settings at the time of Acceptance Testing are given below. As shown, only purging of files is applied (i.e., no folder level actions or archive operations are configured). The key information used for purging operations is provided by the three fields, Source, Pattern and Threshold. The examples below simply state that all files (i.e., pattern is *.*) in each given source folder should be purged after not being updated for more than the number of days indicated by the Threshold field, 30 and 180 days in these two cases.

| Source | InUse | Action | Threshold | Type | Pattern | Dest. |
|-------------------------------------|-------|--------|-----------|-------|---------|-------|
| C:\JRun4\logs | 1 | Purge | 30 | files | *.* | NULL |
| C:\JRun4\servers\IBRD\dispatch\temp | 1 | purge | 180 | files | *.* | NULL |

2.6 Database Archival

2.6.1 IBRD ArcPurgeTables Application

The table archive and purge application (IBRD ArcPurgeTables.exe) takes care of archiving data from various tables in the IBRD database. This program moves “aged out” records to similar tables found in a second database, the IBRD Archive database. The number of days that records are kept in each table in the main IBRD database is configured by setting appropriate values in the SystemCfg table. The tables on the following list are archived in this manner, and example settings for the “age out” criteria are given in the section below. Further details

regarding the underlying logic and algorithms provided in the IBRD Software Maintenance Manual.

OPERMSGLOG – logging for system events and errors
 LOGUSERACCESS – logging of user access
 LOGQUERYACCESS – logging of queries run by users
 LOGPRTEMAILFAX – logging of automatically generated emails
 SARTRANSACTIONLOG – logging of record level activity (e.g., add, update)
 CONFIGCHANGELOG – logging of all changes to the Main table, field by field
 FEEDBACK – holds records generated by users by using the Feedback option
 REGISTRATIONDB406 – the Main table that hold Registration Records

This application is set up to run as a Windows Scheduled Task. To change the scheduled time that the batch program runs, apply similar steps as given in Section 2.4 above.

2.6.2 Example Settings

The following variables are defined in SystemCfg (See Section 3.1 for more information regarding the SystemCfg table).

| SystemCfgName | SystemCfgValue |
|------------------------------|----------------|
| DAYSTOKEEP OPERMSGLOG | 180 |
| DAYSTOKEEP LOGUSERACCESS | 180 |
| DAYSTOKEEP LOGQUERYACCESS | 180 |
| DAYSTOKEEP LOGPRTEMAILFAX | 180 |
| DAYSTOKEEP SARTRANSACTIONLOG | 180 |
| DAYSTOKEEP CONFIGCHANGELOG | 90 |
| DAYSTOKEEP FEEDBACK | 365 |
| DAYSTOKEEP REGISTRATIONDB406 | 3650 |

2.7 System Downtime and “User Warning” Function

There are times when the IBRD System must be shutdown to perform maintenance activities, perhaps most noteworthy for software updates as discussed in Section 2.8 just below. The IBRD application includes a facility for warning users that a system downtime is planned so that they know how long they have to complete their current activity. This facility uses settings from the SystemCfg Table (See Section 3.1). The following steps should be followed to invoke the downtime “user warning”:

1. Determine the time the System will be Shutdown and update the parameter named TIMEOFFLINE in SystemCfg accordingly.
2. Determine the length of time in minutes that the System will be down and update the parameter named LENGHTIMEOFFLINE in SystemCfg accordingly.
3. Determine the length of time in minutes before the planned downtime that the warning should start appearing on the screen, and update the parameter named WARNINGTIMEOFFLINE in SystemCfg accordingly.

4. Set the parameter named **SYSTEMOFFLINE** in SystemCfg to the value 'Y', to effectively activate the "user warning" actions. (Be sure to set it back to 'N', once the System is taken down).

2.8 Installing Software Updates

The above steps in Section 2.7 should be performed to provide an appropriate warning to users that the system will be down for maintenance. The following general steps should be followed to install new code releases:

1. Identify all release files and destination folders as included in associated update instructions.
2. Stop the JRun IBRD Server: access the Windows "Services" dialog box and right click the "JRun IBRD Server" on the list selecting "Stop".
3. Go to the IBRD Server folder (e.g., D:\JRun4\servers\IBRD\) and/or others as indicated and make copies of all existing files to be replaced (as identified in Step 1).
4. Copy the new the files from the CD into the designated folders.
5. Restart the IBRD Server: access the Windows "Services" dialog box and right click the IBRD JRun Server on the list selecting "Start".

- END OF SECTION 2 -

page left blank

3. SYSTEM CONFIGURATION

The elements discussed below contain information for the customization of the runtime environment of the main IBRD application. This configuration information predominately involves application specific parameters and other content which is stored in SQL database tables and Properties files. As applicable, other types of configurable information (e.g., text for standard user emails) are discussed here as well.

A Properties file (extension *.properties) is a simple format text file that contains name (or key) and value pairs on each physical line in the file. Specifically, the format of each line takes the form: <name>=<value>. Properties files are used predominately within the IBRD application to maintain on screen information (e.g., labels, error messages etc.) for the multi-lingual capability. These multi-lingual support files are contained in the system build and are not available for dynamic changes to reconfigure the runtime environment. As such they are discussed further in the IBRD Software Maintenance Manual and not here. One Properties file, named DocumentManager.properties, is discussed in some detail below under the Email Support Section.

There are many SQL configuration tables that control the behavior of the IBRD system. The majority of these table store internal values used by the software in association with various “on screen” functions and data elements. These tables predominately provide a mechanism for data lookup in the Web-based application to reduce data entry errors and supply user assistance. Again these underlying application support configuration tables are discussed further in the IBRD Software Maintenance Manual and not here.

Three configuration tables are addressed in this document. Two tables, MidInfoCfg and PocInfoCfg, are involved in logic that determines by country code which beacon owners and/or National Administrations have authority to enter or modify registration records. Specifically, the MidInfoCfg provides the control settings per country code while the PocInfoCfg table maintains the Point of Contact (POC) information for redirecting users to the proper authorities when a given country is not “supported” by the IBRD. The third configuration table addressed in this document and discussed first as follows, is the SystemCfg table which serves as the central table for runtime system parameters.

3.1 SystemCfg Table

3.1.1 Purpose

The purpose of the SystemCfg table is to store the IBRD System configuration data. Key values stored in this table include the email address of the IBRD POC (i.e., Database Administrator), parameters associated with the automatic generation of “requests for confirmation” and various age out criteria for data archival.

3.1.2 Table Layout

| Field Name | Type | Bytes | Description |
|----------------|---------|-------|-----------------------------------|
| SystemCfgName | varchar | 30 | Name of the value |
| SystemCfgValue | varchar | 100 | The current setting for the value |

Primary Key – SystemCfgName

3.1.3 Example Settings

Example settings for SystemCfg (as used in initial deployment for Acceptance Testing):

| SystemCfgName | SystemCfgValue |
|------------------------------|-------------------------------|
| CONFIRMATION REQUEST TIME | 60 |
| IBRD POC EMAIL | ibrd@noaa.gov |
| DB ADMIN DEFAULT PASSWORD | ?ibrdpass1? |
| DB ADMIN SECURITY ANSWER | Sparky |
| DB ADMIN SECURITY QUESTION | What is the name of your pet? |
| SYSTEMOFFLINE | N |
| TIMEOFFLINE | 15:05 |
| LENGTHTIMEOFFLINE | 5 |
| WARNINGTIMEOFFLINE | 5 |
| IBRD URL | www.cospas-sarsat.ibrd.org |
| DAYSTOKEEP OPERMSGLOG | 180 |
| DAYSTOKEEP LOGUSERACCESS | 180 |
| DAYSTOKEEP LOGQUERYACCESS | 180 |
| DAYSTOKEEP LOGPRTEMAILFAX | 180 |
| DAYSTOKEEP SARTRANSACTIONLOG | 180 |
| DAYSTOKEEP CONFIGCHANGELOG | 90 |
| DAYSTOKEEP FEEDBACK | 365 |
| DAYSTOKEEP REGISTRATIONDB406 | 3650 |

3.2 MidInfoCfg Table

3.2.1 Purpose

The MidInfoCfg table provides storage for Maritime Identification Digits (MID) or “country code numbers” associated with countries around the world. Each row stores the corresponding country name along with several important processing related parameters.

Specifically, the “Supported” field is used to determine whether or not an individual beacon owner may register a beacon in the IBRD. If the value is set to ‘N’, the user will be provided with the proper Point of Contact (POC) for a beacon encoded with the given country code. This POC information is found by following the corresponding PocIndex cross reference entry to the associated PocIndex in the PocInfoCfg table.

This “Supported” flag also excludes beacon owners from using the IBRD where the National Registration database is maintained elsewhere (e.g., USA, France, Canada etc.). Stated another way, National Data Providers are explicitly able to register beacons with the given country code, but individual beacon owners may not.

The ConfirmationRequired field allows for a National Data Provider, or perhaps by National decision even if beacons are registered directly, to identify whether or not “requests for confirmation” should be sent to associated individual beacon owners. Specifically, these requests are sent to the email address in the database for the given beacon record.

3.2.2 Table Layout

| Field Name | Type | Bytes | Description |
|----------------------|----------|-------|---|
| Mid | smallint | 2 | Country identifier (Maritime Identification Digits) |
| CountryName | varchar | 16 | Country Name |
| PocIndex | smallint | 2 | Cross Reference into RegPocCfg Table |
| Supported | char | 1 | Individual Beacon Owners may register - Values: Y, N |
| ConfirmationRequired | char | 1 | Confirmation is required for a particular country Values: Y, N |

Primary Key – Mid

Foreign Key – PocIndex

3.3 PocInfoCfg Table

3.3.1 Purpose

The PocInfoCfg table stores Point of Contact (POC) information for beacon registrars in various countries. The central purpose is to provide individual beacon owners with information for beacons with country codes may not be directly registered in the IBRD system (See discussion above regarding MidInfoCfg for more information).

3.3.2 Table Layout

| Field Name | Type | Bytes | Description |
|---------------------|---------------------|-------|--|
| PocIndex | smallint | 2 | Unique identifier for record |
| BcnRegPocName | varchar nullable | 48 | Beacon Registration Point of Contact name (www.cospas-sarsat.org) |
| BcnRegTelexCntryCd | varchar nullable | 4 | Beacon Registration Point of Contact Telex number (www.cospas-sarsat.org) |
| BcnRegTelexNum | varchar nullable | 16 | Beacon Registration Point of Contact Telex number (www.cospas-sarsat.org) |
| BcnRegTelexAnswerBk | varchar nullable | 16 | Beacon Registration Point of Contact Telex Answer Back code (www.cospas-sarsat.org) |
| BcnRegFAX | varchar nullable | 16 | Beacon Registration Point of Contact Facsimile number (www.cospas-sarsat.org) |
| BcnRegAFTN | varchar nullable | 16 | Beacon Registration Point of Contact AFTN code (www.cospas-sarsat.org) |
| BcnRegPhoneCntryCd | varchar nullable | 5 | Beacon Registration Point of Contact Telephone Country Code (www.cospas- sarsat.org) |
| BcnRegPhoneNum | varchar nullable | 16 | Beacon Registration Point of Contact Telephone Number (www.cospas-sarsat.org) |
| BcnRegMaintName | varchar nullable | 128 | Name of organization that maintains Beacon Registrations (www.cospas-sarsat.org) |
| BcnRegMaintAddress | varchar nullable | 128 | Address of organization that maintains Beacon Registrations (www.cospas-sarsat.org) |
| BcnRegEmail | varchar nullable | 48 | Point-Of-Contact Email Address |

Primary Key – PocIndex

3.4 Email Support

The IBRD system will automatically generate email messages to users. Several aspects of these automatically generated email messages may be configured at runtime. An example of the DocumentManager.properties file, which provides basic control of email parameters such as the name of the resident email server, is given below. Further below, the files that provide the text body of these automatically generated emails are discussed.

3.4.1 DocumentManager Properties File

DocumentManager.properties resides in the IBRD JRun Server folder (e.g., D:\JRun4\Servers\IBRD) and can be used to dynamically configure the system at runtime. An example DocumentManager.properties file (as of initial deployment for Acceptance Testing) is given as follows:

```
#####
#####
##
## NOTE: ANT scripts replace the KEYWORDS such as the root directory. Do not
##       change the KEYWORD name.
##
#####
#####

PrintDebug.printMessagesFlag=true
PrintDebug.logFileName=E:/jrun4/servers/IBRD/dispatch/temp/printDebugLog.txt
RuntimeExecProcessor.consumeBuffers=false

#####
#
# DocumentManager Properties
#
#####

# This boolean flag determines if the temporary files should
# be removed after they are dispatched. A setting of "false"
# leaves the file in the below directory
DocumentManager.removeDocumentsAfterUse=false

# This directory hosts the temporary files.
# IMPORTANT: THIS SHOULD HAVE ALL DOUBLE REVERSE SLASHES
# Example: c:\\jrun4\\servers\\default\\IBRD-ear\\dispatch\\temp
DocumentManager.documentDirectory=E:\\jrun4\\servers\\IBRD\\dispatch\\temp

# DO NOT change this property value
DocumentManager.simulatedNormalMode=false

#####
#
# EmailDispatcher Properties
#
#####

# A common name to easily identify the email server (used only for logging)
EmailDispatcher.commonName=IBRDEmailServer1

# Host server name (network id)
EmailDispatcher.smtpHost=NotReadyYet.nes3.nesdis.noaa.gov

# Email address to be used as "from" address for all the emails
EmailDispatcher.fromAddress=xxxbeaconregistration@noaa.gov

# Flag to use the real email address on Beacon Account or the test address (see below)
# as "to" address. testaddress is effective only when the flag is set to false
EmailDispatcher.sendToRealUser=true
# EmailDispatcher.testAddress=IBRD@noaa.gov

#####
#
# DocumentAuthor Properties
#
#       Properties point to other files containing content for letters
#
#####

# Request for Confirmation Letter templates
DocumentAuthor.ConfReqEmailFileEnglish=E:/jrun4/servers/IBRD/dispatch/templates/confir
mation_request_email_English.txt
DocumentAuthor.ConfReqEmailFileFrench=E:/jrun4/servers/IBRD/dispatch/templates/confir
mation_request_email_French.txt
DocumentAuthor.ConfReqEmailFileRussian=E:/jrun4/servers/IBRD/dispatch/templates/confir
mation_request_email_Russian.txt
DocumentAuthor.ConfReqEmailFileSpanish=E:/jrun4/servers/IBRD/dispatch/templates/confir
mation_request_email_Spanish.txt
# Registration "Information Provided" Letter templates
DocumentAuthor.RegEmailFileEnglish=E:/jrun4/servers/IBRD/dispatch/templates/registrati
on_email_English.txt
```

```
DocumentAuthor.RegEmailFileFrench=E:/jrun4/servers/IBRD/dispatch/templates/registratio
n_email_French.txt
DocumentAuthor.RegEmailFileRussian=E:/jrun4/servers/IBRD/dispatch/templates/registrati
on_email_Russian.txt
DocumentAuthor.RegEmailFileSpanish=E:/jrun4/servers/IBRD/dispatch/templates/registrati
on_email_Spanish.txt
```

3.4.2 Email “Text Body” Files

The email “text body” files are simple text files that contain the content of the letters to be sent to end users. Clearly, only users who provide an email address (i.e., the beacon record has a valid email address) will be sent messages. There are two general formats. One format is sent after a user enters a new registration or updates an existing one. The purpose of this letter is to provide the data just stored in the database to the user so that the information can be verified. The second email format requests a user to confirm and/or update the information stored on line (See Section 2.4 regarding the Request for Confirmation Process for more information).

There are actually four files for each format, one for each of the languages supported by the IBRD: English, French, Russian and Spanish. The files names and locations are stored in the DocumentManager.properties file discussed just above. The English version of each letter, as configured for Acceptance Testing, is listed as follows. It should be noted that the salutation line (i.e., Dear <beacon owner>), as well as the URL (found in SystemCfg) are added by the software, at the beginning and ending respectively, when the email is generated. Finally, each email is accompanied by an attachment consisting of a simple field by field text listing of the registration information currently on record in the database.

Example of registration_email_English.txt:

```
Thank you for providing your registration information. This information has been
shown to be a great benefit to SAR forces in responding to beacon activations and
significantly reduces their response time. Periodic confirmation of this data will
be requested every two years. If you have any questions regarding your
registration, please contact us via email or by using the feedback survey on the
IBRD web site.
```

```
Sincerely,
IBRD Database Administrator
```

Example of confirmation_request_email_English.txt:

```
It has been almost two years since you provided new registration information for
your 406 MHz emergency beacon. Please go to the Cospas-Sarsat IBRD web site and
review your registration to ensure that all of the information is current.
Specifically, please do one of the following:
```

```
If you need to make changes or update any information, please do so accordingly.
```

```
If no changes are required, proceed directly to the "Acknowledge Confirmation
Request" Page and select the "Confirm" button to acknowledge this request for
confirmation and that existing registration data is correct;
```

```
If the status of your beacon has changed (e.g. replaced, sold), please go to the
"Change Beacon Status" Page and mark the appropriate status, providing comments as
necessary.
```

```
Registration information has been shown to be of great benefit to Search and
Rescue (SAR) forces in responding to beacon activations and significantly reduces
response time. In addition to keeping your registration information up-to-date, it
is imperative that you refer to the owner's manual for your beacon and perform any
```

required maintenance or testing. The beacon's battery must be replaced periodically. For Category 1 EPIRBs (automatic activation), the hydrostatic release mechanism requires periodic replacement.

If you have any questions regarding your registration, please contact us via email or by using the feedback survey on the IBRD web site.

Sincerely,
IBRD Database Administrator

- END OF SECTION 3 -

page left blank

4. LOGGING

There are two general methodologies used by the main IBRD application to record or “log” system events. These are SQL database tables and text files. The text files, although briefly identified and described further below in this section, are aimed more at the application software maintenance personnel. The various logging tables, while useful at both levels are aimed more at system level maintenance.

4.1 ConfigChangeLog Table

4.1.1 Purpose

The purpose of the ConfigChangeLog table is to store each and every change made to individual fields within records of the main table. (The main table is named RegistrationDb406, and is documented in the IBRD Software Maintenance Manual). The ConfigChangeLog table provides a powerful tool for analyzing system problems. Also, although it is hoped that it would never be necessary, this mechanism provides the means to reconstruct any record in the database should records in the main table somehow become corrupted. The schema is taken from a more general purpose table used in the USA for various applications and hence some fields remain that are not really needed or used in this system.

4.1.2 Table Layout

| Field Name | Type | Bytes | Description |
|-------------|-----------------|-------|--|
| ChangeId | int Identity | 4 | Configuration Change Id |
| ChangeIdLnk | int | 4 | Configuration Change Id Link: the ChangeId of the first change in a group of related changes. 0 means that the change is not part of a group. (SQL Default is 0) |
| OprMsgId | int | 4 | Unique Id for associated Operator Message, if any. (SQL Default is 0) |
| ChangeTime | datetime | 8 | Time of Configuration change |
| DBLevel | varchar | 1 | (Highest) Database level of the Configuration change: ‘R’ = Entire Row ‘F’ = Field within Row |
| ChangeType | varchar | 1 | Type of Configuration change: ‘A’ = add item ‘U’ = update item |

| Field Name | Type | Bytes | Description |
|-------------|---------------------|-------|---|
| TempFlag | varchar | 1 | Configuration change is temporary flag: 'Y' = yes, expect change to be reversed 'N' = no, expect change to be permanent |
| TableName | varchar nullable | 32 | Name of Table changed |
| RowId | varchar nullable | 255 | Identifier (Primary Key) of Row in Configuration Table changed. |
| FieldName | varchar nullable | 64 | Name of Field in Configuration Table changed. If null, the change applies to the entire row. |
| OldValue | varchar nullable | 255 | Old (previous) value of changed field |
| NewValue | varchar nullable | 255 | New value of changed field |
| Program | varchar | 16 | Program name |
| UserId | varchar | 16 | User identifier – See field OperatorId in Registration406Db (main table) |
| SubsysId | varchar | 4 | (Always set to 'IBRD') |
| TrackingId | varchar nullable | 16 | External tracking identifier (always NULL for IBRD) |
| Remarks | varchar | 255 | User remarks (e.g., the reason that the change was made) |
| SessionId | varchar | 50 | Session Identifier |
| SessionDate | datetime | 8 | Date/Time of Session log |

Primary Key – ChangeId

Index1 – ChangeTime

Index2 – SessionId

4.2 OperMsgLog Table

4.2.1 Purpose

The purpose of the OperMsgLog table is to store major processing related events and/or errors. While this table can provide useful information, the implementation at the time of Acceptance Testing relies more on other tables and text files for much of the logging originally expected to be stored here. The schema is taken from a more general purpose table used in the USA for various applications and hence some fields remain that are not really needed or used in this system.

4.2.2 Table Layout

| Field Name | Type | Bytes | Description |
|--------------|-------------------|-------|---|
| OprMsgId | int Identity | 4 | Unique Id for Operator Message |
| EventTime | datetime | 8 | Time of event |
| OprAckTime | datetime nullable | 8 | Time acknowledged by the Operator |
| ProgPrior | tinyint | 1 | Programmer priority |
| OperPrior | tinyint | 1 | Operator priority 00 - 09 Not used 10 - 19 Analyst Level (not for operator) 20 - 29 Operator Info (no action required) 30 - 39 Operator Action (action required) 40 - 49 Emergency (Immediate action req.) |
| OperDisplay | varchar | 1 | Operator Display requested: 'Y' = Yes (generic display), 'N' = No (SQL Default is 'N') |
| SubsysId | varchar | 4 | (Always set to 'IBRD') |
| SubsysMsgNum | int | 4 | Message number assigned by the Subsystem that generated the message |
| Program | varchar | 16 | Program name |
| UserId | varchar | 16 | User identifier creating the message |
| UserIdAck | varchar nullable | 16 | User identifier acknowledging message ("AckRequired" indicates message requires and has not received acknowledgement) |
| Message | varchar | 255 | Text message generated by the Subsystem |
| TableName | varchar nullable | 32 | Name of primary table for related data |
| TableId | int | 4 | Unique Id in table for related data (SQL Default is 0) |

Primary Key – OprMsgId**Index1** – EventTime, SubsysId, SubsysMsgNum

4.3 LogUserAccess Table

4.3.1 Purpose

The purpose of the LogUserAccess table is to store information for all access, successful or failed, to the IBRD user interface. Beyond the basic recording of who is logging in and out, this table can be used to track possible abusive users, to provide some measure of software performance and to address related security issues. This table is a good place to obtain system usage statistics.

4.3.2 Table Layout

| Field Name | Type | Bytes | Description |
|--------------|---------------------|-------|--|
| LogId | int Identity | 4 | Log Identifier |
| BeaconId | varchar nullable | 15 | 406 Beacon Id code |
| UserName | varchar nullable | 16 | User Name – See field OperatorId in Registration406Db (main table) |
| OwnerName | varchar nullable | 48 | Owners name, last name first, name of company or name of federal or state agency followed by application |
| Password | varchar | 16 | Password |
| IPAddress | varchar | 20 | IP Address |
| SessionId | varchar | 50 | Session Identifier |
| SessionDate | datetime | 8 | Date/Time of Session log |
| SessionEvent | varchar | 10 | Logs the event that occurred during this session Values: Log In Log Out |

Primary Key – LogId

4.4 LogQueryAccess Table

4.4.1 Purpose

The purpose of the LogQueryAccess table is to store logging information for all query operations. This table can be used to track usage and to provide some measure of software performance.

4.4.2 Table Layout

| Field Name | Type | Bytes | Description |
|---------------|-----------------|-------|--|
| LogId | int Identity | 4 | Log Identifier |
| UserName | varchar | 16 | User Name – See field OperatorId in Registration406Db (main table) |
| Password | varchar | 16 | Password |
| IPAddress | varchar | 20 | IP Address |
| Query | varchar | 2000 | Query |
| NumRecsReturn | int | 4 | Number of records the query returned |
| ExecuteTime | float | 8 | Time for execution (in seconds) |
| SessionId | varchar | 50 | Session Identifier |
| SessionDate | datetime | 8 | Date/Time of Session log |

Primary Key – LogId

4.5 LogPrtEmailFax Table

4.5.1 Purpose

The purpose of the LogPrtEmailFax table is to store logging information related to transactions involving sending email to users. This table will record when registration information or request for confirmations have been produced. (The name of the table involves historic references to Printing (Prt) and Facsimile (Fax) events, but only email is used in the IBRD system for user correspondence).

4.5.2 Table Layout

| Field Name | Type | Bytes | Description |
|---------------|---------------------|-------|--|
| LogId | int Identity | 4 | Log Identifier |
| BeaconId | varchar | 15 | 406 Beacon Id code |
| BeaconRegType | tinyint | 1 | Beacon Registration Type (output format) |
| UserName | varchar nullable | 16 | User Name – See field OperatorId in Registration406Db (main table) |

| Field Name | Type | Bytes | Description |
|---------------|---------------------|-------|---|
| OwnerName | varchar nullable | 48 | Owners name, last name first, name of company or name of federal or state agency followed by application |
| Phone | varchar nullable | 24 | Telephone Number |
| ProcessType | varchar | 5 | Type of Transmission: Always EMAIL |
| ProcessReason | varchar | 30 | Reason for Processing Values: NEW REGISTRATION, INFORMATION UPDATE, TWO YR REQUEST CONFIRM LETTER |
| ProcessDate | datetime | 8 | Date of Processing |
| SessionId | varchar | 50 | Session Identifier |
| SessionDate | datetime | 8 | Date/Time of Session log |

Primary Key – LogId, BeaconId

4.6 Text Based Log Files

The text log files generated directly by the IBRD software reside in the IBRD JRun Logs directory (e.g., D:\JRun4\logs). Separate files are generated for each day. Specifically, given something to enter in the log, and the time has passed midnight on a given day, the file is renamed using the previous day's date and a new file is opened. All IBRD generated text log files are purged on the basis of age by the IBRDFileArcPurge application as discussed above in Section 2.5.

4.6.1 System Events Log

The file named, IBRD-Event.log, generally contains events that occur at the Java software processing level. The depth of information here is relatively limited but it can be used in diagnosing system problems. (There is also a very similar file named noaa.log that essentially contains redundant information and should really have been eliminated in the IBRD development process, but was overlooked).

4.6.2 System Errors Log

The file named, Errors.log, clearly contains system errors. Perhaps the most powerful aspect of this file is that in many cases it will provide a "call stack" listing of software modules referenced leading up to the point of the error. This information can be very useful to programmers diagnosing in diagnosing and correcting problems.

4.6.3 Apache HTTP Server Log

As discussed in Section 2.2 above, Apache is used as the HTTP Server and runs uniquely on the Web Server computer. There is also a log file generated by this third-party software package. This file can be used for debugging general web site access problems, and perhaps of most interest for general web site level statistics.

- END OF SECTION 4 -

page left blank

5. OTHER SUPPORTING TABLES

There are several other tables listed in the subsections below that are of interest to system operations, and perhaps in particular the Database Administrator. Additional tables mentioned below that pertain to underlying configuration settings and/or details are generally found in the IBRD Software Maintenance Manual.

5.1 Users Table

5.1.1 Purpose

The purpose of the Users table is to store user account information. Users with accounts are National Data Providers, Ship Surveyors, Search and Rescue (SAR) services and Database Administrator(s). These types of users generally perform functions involving multiple beacons. Different functions and information will be available depending on the access role (RoleId) assigned. The details of the various user functions may be found in the Functional Requirements and User Interface Design documents. Modifications to this table are performed online by Database Administrators by accessing the appropriate link after logging into the IBRD System.

5.1.2 Table Layout

| Field Name | Type | Bytes | Description |
|------------|---------------------|-------|---|
| UserName | varchar | 16 | User Name (Logon Id) |
| Password | varchar | 16 | User password |
| RoleId | int | 4 | Role Identifier (See RolesCfg) |
| Status | varchar | 1 | User account Status (See RecordStatusCfg) |
| FirstName | varchar | 20 | User First Name |
| LastName | varchar | 30 | User Last Name |
| OrgName | varchar nullable | 50 | Organization Name |
| Title | varchar nullable | 20 | User's Title |
| Phone | varchar nullable | 24 | User Telephone Number |
| Email | varchar nullable | 48 | User Email address |

| Field Name | Type | Bytes | Description |
|-------------------|---------------------|-------|--|
| CountryNumber | varchar nullable | 32 | For National Data Providers: the country number value(s) associated with / assigned to this nation separated by semi-colons (See CountryCfg) |
| ChallengeQuestion | varchar nullable | 64 | Password Challenge Question Selected for supporting re-instatement of an Account Password |
| ChallengeResponse | varchar nullable | 24 | Password Challenge Response to the Password Challenge Question |

Primary Key – UserName

5.2 Feedback Table

5.2.1 Purpose

The purpose of the Feedback table is to store user-entered feedback information, by using the available link on the IBRD online interface. The feedback mechanism is provided to allow users to report system problems, recommend improvements and/or provide general information.

5.2.2 Table Layout

| Field Name | Type | Bytes | Description |
|------------|---------------------|-------|---------------------|
| FeedbackId | int Identity | 4 | Feedback Identifier |
| UserType | varchar | 30 | User Type or Role |
| FirstName | varchar nullable | 20 | User First Name |
| LastName | varchar nullable | 30 | User Last Name |
| OrgName | varchar nullable | 50 | Organization Name |
| Address | varchar nullable | 48 | User Address |
| City | varchar nullable | 32 | User City |
| State | varchar nullable | 2 | User State |

| Field Name | Type | Bytes | Description |
|----------------|---------------------|-------|--|
| ZipCode | varchar nullable | 9 | User Zip Code |
| Email | varchar nullable | 48 | User Email address |
| Phone | varchar nullable | 24 | User Telephone Number |
| PhoneType | varchar nullable | 4 | Type of Telephone Number Values: HOME, WORK, CELL, FAX, OTHR |
| Description | varchar nullable | 2000 | Description of Feedback (includes problems encountered, improvements/enhancements) |
| Action | varchar | 30 | Action performed Values: VIEWING, MODIFICATION, QUERIES |
| FeedbackReason | varchar | 16 | Feedback Reason Values: COMMENT, SYSTEM PROBLEM, REG PROBLEM, ENHANCEMENT |
| FeedbackDate | datetime | 8 | Date of Feedback |
| ResponseDate | datetime | 8 | Date that Feedback was responded to (if any) by the Database Administrator (manually managed by same). |

Primary Key – FeedbackId

Index – FeedbackDate

5.3 Reporting Tables

5.3.1 Purpose

The purpose of the Reporting Tables is to allow for the production of monthly and annual statistics. The reports may also be generated with specific user provided start and end times. There are several tables that provide for a user interface based reporting capability. Although it is provided, this capability is somewhat limited and only provides for a basic level of “canned” reports. The design here uses “dimension” tables to identify various criteria that will be stored in the table named BeaconReportFact. The purpose or intended advantage of this approach is that it prevents reports being run against the main table itself, and hence having an impact on system performance. Although this design has some merit, it generally only provides a significant advantage with extremely large data

sets. The price for this performance gain is a very high trade off in the lack of flexibility.

It is possible that Cospas-Sarsat will decide to enhance this functionality but it is anticipated that back end operations using database tools such as Microsoft Access or SQL Server will provide much more productive solutions for Cospas-Sarsat needs. Regardless, there are several tables employed by the IBRD system to provide the available reports. As mentioned above the actual data that appears in any given “canned” report is stored accordingly in the BeaconReportFact table. The various underlying dimension tables are listed as follows:

- BeaconManufacturerDim - Beacon Manufacturer Keys and Names
- BeaconRegTypeDim – Registered Beacon Types (EPIRB, ELT, PLB)
- CountTypeDim – types of beacon events that are to be counted
- TimeDim – single day values (see further discussion below)

Beacon country code is also broken out in the reporting functionality, but no dimension table was deemed to be necessary.

Below, only the BeaconReportFact table and TimeDim table are detailed. It is worth noting that the Database Administrator may find the BeaconReportFact table itself useful for back end analysis operations.

It is more important to note that the TimeDim table does require maintenance. The purpose of the TimeDim table is to store the “time dimension” or “days of the year”. An original USA design intention that all events should be identified on a “daily” basis generated this need for specific date column in BeaconReportFact. The software will not function properly if the “current day” is not in this table and hence this table **must** be periodically populated with data pertaining to future dates. The IBRD system has been delivered with these dates populated out to 31 December 2014, so this situation is not a pressing matter, but it can not be simply ignored either.

5.3.2 BeaconReportFact Table Layout

| Field Name | Type | Bytes | Description |
|-----------------------|------|---------|------------------------------|
| TimeKey | 4 | int | Time Key |
| BeaconRegTypeKey | 1 | tinyint | Beacon Registration Type Key |
| BeaconManufacturerKey | 4 | int | Beacon Manufacturer Key |
| CountTypeKey | 4 | int | Count Type Key |
| Count | 4 | int | Count of a Beacon Event |
| BeaconCountryCode | 4 | int | Encoded Beacon Country Code |

Primary Key – TimeKey

5.3.3 TimeDim Table Layout

| Field Name | Type | Bytes | Description |
|---------------|----------|-------|---|
| TimeKey | int | 4 | Time Key |
| CalendarYear | varchar | 4 | Calendar Year |
| CalendarMonth | varchar | 10 | Values: January, February, March, April, May, June, July, August, September, October, November, December |
| CalendarDay | smallint | 2 | Values: 1 through 31 |
| DayOfWeek | varchar | 10 | Values: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday |
| CalendarDate | datetime | 8 | Calendar Date |

Primary Key – TimeKey

- END OF SECTION 5 -

page left blank

ANNEX A**LIST OF ABBREVIATIONS AND ACRONYMS**

| | |
|-----------|---|
| COSPAS | Space system for the rescue of vessels in distress (Russian Federation) |
| COTS | Commercial off the shelf |
| C/S | Cospas-Sarsat |
| ELT | Emergency Locator Transmitter |
| EPIRB | Emergency Position Indicating Radio Beacon |
| FAX | Facsimile |
| FAQ | Frequently asked question |
| IBRD | International 406 MHz Beacon Registration Database |
| ICAO | International Civil Aviation Organization |
| ID | Identification |
| IMO | International Maritime Organization |
| IP | Internet protocol |
| ITU | International Telecommunication Union |
| MARS | Maritime mobile access and retrieval system (ITU database) |
| MCC | Cospas-Sarsat Mission Control Centre |
| MHz | Megahertz |
| MMSI | Maritime Mobile Station Identity |
| POC | Point of contact |
| PLB | Personal Locator Beacon |
| RCC | Rescue Coordination Centre |
| SAR | Search and Rescue |
| SARSAT | Search and Rescue Satellite Aided Tracking system (Canada, France, USA) |
| SBM | Shore-based maintenance |
| SQL | Structured query language |
| TAC | Type approval certificate |
| 15 Hex ID | 15 hexadecimal character identification |

- END OF ANNEX A -

- END OF DOCUMENT -

page left blank

Cospas-Sarsat Secretariat
700 de la Gauchetière West, Suite 2450, Montreal (Quebec) H3B 5M2 Canada
Telephone: +1 514 954 6761 Fax: +1 514 954 6750
Email: mail@cospas-sarsat.int
Website: <http://www.cospas-sarsat.org>
