



RESOURCE AND PATIENT MANAGEMENT SYSTEM

IHS USER SECURITY AUDIT Application Programming Interface

(BUSIA)

Technical Manual

Version 1.0 Patch 4
November 2021

Office of Information Technology
Division of Information Technology

Table of Contents

1.0	Introduction.....	1
2.0	Implementation and Maintenance.....	2
2.1	System Requirements	2
2.2	Package-wide Variables	2
2.3	Security Keys.....	2
3.0	Menu Diagram.....	3
4.0	Routine Descriptions	5
4.1	Routine List.....	5
4.2	Routines with Description	5
4.3	BUSA Logging Utilities.....	6
4.3.1	RPMS Logging API – \$\$LOG^BUSAAPI	6
4.3.2	RPMS Logging API - \$\$BYPLOG^BUSAAPI	7
4.3.3	RPMS Audit File API - \$\$FAUD^BUSAAP	7
4.3.4	RPMS Logging API - FMENT^BUSAAPI	8
4.3.5	RPMS Remediation Entry API – \$\$CHECK^BUSAAPIR.....	8
4.3.6	CIA BROKER/BMXNet/XWB BROKER RPC Logging.....	8
5.0	Files and Tables.....	20
5.1	File List	20
5.2	File Access	20
5.3	Cross References.....	21
5.4	Table File.....	23
6.0	External Relations	28
6.1	Callable Routines.....	28
6.2	Published Entry Points.....	28
6.3	Exported Options	28
7.0	Internal Relations	30
8.0	Archiving and Purging	31
8.1	Purging functionality included with the Patch 4 release	31
8.2	Archiving and Purging Functionality Included with the Patch 3 Release	31
9.0	Documentation Resources	32
9.1	System Documentation.....	32
9.2	%INDEX	32
9.3	List File Attributes Option.....	32
10.0	SAC Requirements and Exemptions	34
11.0	Templates, Forms, and Protocols	35
11.1	Print Templates.....	35

- 11.2 Sort Templates 35
- 11.3 Input Templates 35
- 11.4 List Templates 35
- 11.5 Forms 35
- 11.6 Protocols..... 35
- 12.0 Sample BUSA Audit RPC Definitions 36**
 - 12.1 BEHOENCV DETAIL 36
 - 12.2 BEHOENCX VISITLST 36
 - 12.3 BEHOENCV LIST 36
 - 12.4 BEHOENCX FETCH..... 36
 - 12.5 BEHOENCX UPDPRV 36
 - 12.6 BEHOENCX PTINFO 37
 - 12.7 BGOVPOV SET 37
 - 12.8 BGOVPOV GET 37
 - 12.9 BGOPROB SET 37
 - 12.10 BQI GET DX CAT BY PANEL..... 37
 - 12.11 BQI LOOKUP PATIENTS 38
 - 12.12 BJPN UPDATE PROBLEM 38
 - 12.13 BJPN UPDATE PROBLEM 38
- 13.0 Ensemble Classes Used by BUSA 39**
 - 13.1 BUSA Report Classes 39
 - 13.2 BUSA Remediation Classes 39
- Glossary 41**
- Acronym List 42**
- Contact Information 43**

Document Revision History

Version	Date	Author	Description	Sections
1.0	8/2019	GDIT SESS team	Initial Release	All
1.1	7/2020	Skip Squires	Update for 2015 CHIT Certification Release Patch 3	All
1.2	8/2021	Brian Everett	Update for Patch 4 Release	All

Preface

The purpose of this manual is to provide technical information about the Indian Health Service (IHS) USER SECURITY AUDIT (BUSA) package. The BUSA package contains a number of utilities consisting of application programming interface (API) calls, programming hooks, and remote procedure calls (RPCs) that have been developed to allow user activity to be tracked and reported upon.

Patch 1 contained changes to BUSA to optionally allow a hash value to be stored on saved BUSA summary file entries to allow sites to determine whether the entries have been tampered with. In addition, a new BUSA reporting utility was provided to allow reporting on specific information from Electronic Prescribing of Controlled Substances (ECPS).

Patch 2 contained changes to allow the BUSA Zen reports to display after upgrading to HealthShare.

Patch 3 contained changes to BUSA to add functionality to satisfy the 2015 Certified Health Information Technology (CHIT) Certification (d)(2) and (d)(3) criteria. In addition, new BUSA archive functionality was provided to allow sites to archive their older BUSA audit data.

Patch 4 contains new functionality to allow errors on some reports (currently the EPCS Incident Reports) to be remediated so that they no longer show up on the reports.

1.0 Introduction

The BUSA package is a component of the IHS RPMS that provides sites with the ability to track user activity. This activity can then be reported on to meet meaningful use certification requirements.

This manual provides IHS developers with a technical description of the BUSA APIs, routines, files, menus, cross references, globals, and other necessary information required to properly set up applications to log user activity in BUSA files.

All APIs, routines, files, options, and keys are namespaced starting with the letters BUSA. The file number range for this package is 9002319–9002319.99.

2.0 Implementation and Maintenance

The BUSA utilities are designed to give application developers the ability to track user activity in a consistent manner, while keeping application code changes to a minimum.

2.1 System Requirements

The following table shows the versions of other packages that should be installed for BUSA to work properly.

Table 2-1: Required packages and versions

Module	Minimum Version	Recommended Version
BUSA	v1.0 Patch 3	
Health Share	v2017.2.2	
VA FileMan (DI)	v22.0 Patch 1020	
IHS Kernel Toolkit (XT)	v7.3 through Patch 1019	
VA Kernel (XU)	v8.0 Patch 1020	
BMW	v2020.3	
EHR	v1.1 Patch 31	

2.2 Package-wide Variables

There are no package wide BUSA variables in RPMS.

2.3 Security Keys

The following security keys can be assigned to users.

Table 2-2: Security keys and descriptions

Key Name	Description
BUSAZMGR	This security key should only be assigned to those persons who will manage the BUSA application. It should not be given to the general RPMS user population.
BUSAZRPT	This key should be given to users who require access to the BUSA web-based reporting tool.
BUSAZARCHIVE	This key should only be assigned to those persons who will manage the BUSA application data archive options.
BUSAZREMEDATION	This key should only be assigned to those persons who will be remediating record information.

3.0 Menu Diagram

RPMS menus in the BUSA system:

- **BUSA Main Menu [BUSAMENU]**

This BUSA option allows the user to access the archive functionality and manage and configuration for IHS User Security Audit.

This option requires the BUSAZMGR security key.

```

*****
**   BUSA Main Menu   **
*****
IHS USER SECURITY AUDIT Version 1
2020 DEMO HOSPITAL

BA   BUSA Archive menu...
BS   Edit Security Audit

Select BUSA Main Menu Option:
    
```

Figure 3-1: BUSA Main Menu for IHS User Security Audit

- **Edit Security Audit [BUSA SECURITY EDIT]**

This BUSA option allows the security auditing logging to be turned on or off.

This option requires the BUSAZMGR security key.

- **BUSA Archive Menu ... [BUSA ARCHIVE MENU]**

The BUSA archive main menu option allows the user to access the archive options.

This option requires the BUSAZARCHIVE security key.

```

*****
**   BUSA Main Menu   **
*****
IHS USER SECURITY AUDIT Version 1
2020 DEMO HOSPITAL

AI   Archive BUSA Information
VA   Verify Archive
PA   Purge BUSA Records
LA   Load BUSA Archive File
RR   Remove Restored Records
AR   Archive Report

Select BUSA Archive Menu Option:
    
```

Figure 3-2: BUSA Data Archive menu for IHS User Security Audit

- **Archive BUSA Information [BUSA ARCHIVE BUSA INFORMATION]**

This BUSA option allows the user to archive audit data to external files to alleviate space issues on the RPMS database.

This option requires the BUSAZARCHIVE security key.

- **Verify Archive [BUSA VERIFY ARCHIVE]**

This BUSA option allows the user to verify whether the created archive files are properly formatted and uncorrupted.

This option requires the BUSAZARCHIVE security key.

- **Purge BUSA Records [BUSA PURGE BUSA RECORDS]**

This BUSA archive menu option allows the user to purge audit information from the **BUSA AUDIT LOG SUMMARY** and **BUSA AUDIT LOG DETAIL** files that have been archived into one or more files and verified for accuracy.

This option requires the BUSAZARCHIVE security key.

- **Load BUSA Archive File [BUSA LOAD BUSA ARCHIVE FILE]**

This BUSA archive menu option allows the user to load one or more archive files into the **BUSA AUDIT LOG SUMMARY ARCHIVE** and **BUSA AUDIT LOG DETAIL ARCHIVE** files for review in the Archive Zen report.

This option requires the BUSAZARCHIVE security key.

- **Remove Restored Records [BUSA REMOVE RESTORED RECORDS]**

This BUSA archive menu option allows the user to purge one or more archive files out of the **BUSA AUDIT LOG SUMMARY ARCHIVE** and **BUSA AUDIT LOG DETAIL ARCHIVE** files used for Archive Zen report review.

This option requires the BUSAZARCHIVE security key.

- **Archive Report [BUSA ARCHIVE REPORT]**

This BUSA archive menu option allows the user to generate a report on the archive activity for the RPMS site.

This option requires the BUSAZARCHIVE security key.

4.0 Routine Descriptions

4.1 Routine List

Table 4-1 lists all BUSA routines.

Table 4-1: BUSA routines

Routine	Routine	Routine
BUSA1P01	BUSA1P02	BUSA1P03
BUSA1P04	BUSA1PRE	BUSAACVR
BUSAAPI	BUSAAPIR	BUSAARAI
BUSAARLA	BUSAARO	BUSAARPA
BUSAARPU	BUSAARRP	BUSAARST
BUSAARVA	BUSAAUT	BUSABQI
BUSACLAS	BUSAMAG1	BUSAOPT
BUSARPC	BUSASWCH	BUSATRAN
BUSAUTIL	BUSAUTL1	

4.2 Routines with Description

Table 4-2 lists and describes each routine found in the BUSA release up to and including this patch release.

Table 4-2: Routines and descriptions

Routine	Description
BUSA1P01	Patch 1 post-installation routine
BUSA1P02	Patch 2 post-installation routine
BUSA1P03	Patch 3 post-installation routine
BUSA1P04	Patch 4 post-installation routine
BUSA1PRE	Pre/post-installation routine for BUSA v1.0
BUSAACVR	User access checks for the BUSA Audit Report Utility
BUSAAPI	Main API front end routine
BUSAAPIR	Remediation API routine
BUSAARAI	BUSA Archive BUSA Information option
BUSAARLA	Load BUSA Archive file option
BUSAARO	BUSA Archive Menu options
BUSAARPA	Remove Restored Records option
BUSAARPU	Purge BUSA Records option

Routine	Description
BUSAARRP	Archive Report option
BUSAARST	Archive utility routine (displays)
BUSAARVA	BUSA archive verification
BUSAAUT	BUSA Archive Utility Calls
BUSABQI	iCare specific routine
BUSACLAS	Transport routine for BUSA classes
BUSAMAG1	VistA Imaging specific routine
BUSAOPT	Security audit option routine
BUSARPC	Routine containing hooks for BMXNet, XWB Broker, and CIA Broker calls
BUSASWCH	Turn on/off FileMan Auditing
BUSATRAN	BUSA RPC definition routine
BUSAUTIL	Utility function calls
BUSAUTL1	IHS USER SECURITY AUDIT Utility Program 1

4.3 BUSA Logging Utilities

4.3.1 RPMS Logging API – \$\$LOG^BUSAAPI

This is the primary API for logging information to BUSA. Information will only be logged to BUSA if the Master Switch in the **BUSA SECURITY EDIT** option is set to **ON**. Table 4-3 describes the input parameters expected by this API call.

Table 4-3: Input parameters and descriptions

Parameter	Data Type	Description
TYPE (Optional)	Set of Codes	The type of entry to log (R: RPC Call; W: Web Service Call; A: API Call; O: Other) – Default to 'A'
CAT (Required)	Set of Codes	The category of the event to log (S: System Event; P: Patient Related; D: Definition Change; O: Other Event)
ACTION (Required for CAT = "P")	Set of Codes	The action of the event to log (A: Additions; D: Deletions; Q: Queries; P: Print; E: Changes; C: Copy; AC: Access to patient information; EA: Emergency access to patient information; U: Change to user privilege; AL: Change to audit log status; ES: Change to encryption status)
CALL (Required)	Free Text	Free text entry describing the call which originated the audit request (Maximum length 200 characters). Examples could be an RPC value or calling routine.
DESC (Required)	Free Text	Free text entry describing the call action (Maximum length 250 characters). Examples could be "Patient demographic update," "Copied iCare panel to clipboard," or "POV Entry."

Parameter	Data Type	Description
BUSADET (Required for CAT = "P")	Free Text	This field should contain the name of the variable array set up with the patients to log. For example, given the variable array used below, pass in "BUSADET" for the value of this parameter. BUSADET(#) = [1] ^ [2] ^ [3] ^ [4] ^ [5] Where: [1] Patient DFN – Will be populated based on Visit IEN if blank [2] Visit IEN (optional, required if DFN is blank) [3] Detail information pertaining to this specific detail entry (optional) [4] New value (optional) [5] Original value (optional)
HASH (optional)	Fixed	Set to a '0' if a HASH value should not be stored for the resulting BUSA summary entry. Set to '1' or to null to record a HASH of the entry
<return value>	String	Result returned as: [1] ^ [2] Values: [1] Status = 1: Call completed successfully; 0: Call failed [2] Error Message (if applicable)

4.3.2 RPMS Logging API - \$\$BYPSLOG^BUSAAPI

This is an alternate API for available for logging information to BUSA. Information will get logged to BUSA regardless of whether the Master Switch in the BUSA SECURITY EDIT option is set to ON or OFF. The input parameters for this API are identical to those described in Section 4.3.1.

4.3.3 RPMS Audit File API - \$\$FAUD^BUSAAP

This API call returns whether a FileMan audit should be performed. If no file number is passed in, only the status of the BUSA FileMan switch will be used in determining whether to audit. If a file number is passed in, checks on both the BUSA FileMan switch status and whether the file is included to be audited in the **BUSA FILEMAN AUDIT INCLUSIONS** file will be performed.

Table 4-4: RPMS Audit File API parameter and description

Parameter	Data Type	Description
BFILE	Numeric	The FileMan file number
<return value>	String	Result returned as: 1: Audit the entry 0: Do not audit the entry

4.3.4 RPMS Logging API - FMENT^BUSAAPI

This API call accepts information from a FileMan audit entry and creates a corresponding BUSA entry for it.

Table 4-5: Logging API and description

Parameter	Data Type	Description
BFILE	Numeric	The FileMan file number
BAIEN	Numeric	The FileMan audit IEN pointer
BCALL	String	The calling routine

4.3.5 RPMS Remediation Entry API – \$\$CHECK^BUSAAPIR

This API returns whether a BUSA Remediation entry has been created for the specified Type (or Other Type) and record.

Table 4-6: Returning API, data type, and descriptions

Parameter	Data Type	Description
Remediation Type	String	The remediation type to check. This type could one from the predefined list (BUSAS Entry – HASH Mismatch, BUSAS Entry – Missing, etc.) or it could be a custom value
Remediation Record	String	The remediation record to look up
<return value>	String	Result returned as: 0 – No remediation entry on file 1 – Submitted type/record has been remediated

4.3.6 CIA BROKER/BMXNet/XWB BROKER RPC Logging

This utility allows events to be logged from RPC calls passed through the CIA, XWB and the BMXNet brokers. The following instructions explain how to set up RPC calls to be logged.

4.3.6.1 Setting Up RPC Calls to be Logged

The first step in setting up RPC calls to be logged through the CIA, XWB, and BMXNet brokers is to identify the list of application RPC calls that need to be logged. As a general guideline, any RPC call that is utilized by EHR or iCare to query (display), add, change, delete, print, or copy patient information should be included in the list to be logged.

Once this list of RPC calls has been identified, they need to be set up in the BUSA AUDIT RPC DEFINITIONS file (#9002319.03) to be logged. This file contains the following fields which can be populated for each entry:

RPC (#.01)

This required field is a pointer to the REMOTE PROCEDURE file (#8994). Select the RPC which should be logged in the CIA, XWB, or BMXNet broker.

CATEGORY (#.02)

This required field should be populated with the category best representing the function of the RPC. For the initial release of the IHS SECURITY AUDIT application as it pertains to meaningful use certification, the focus for this release is to track patient-related events. Therefore, most entries for the CATEGORY field will most likely be set to "P." The following codes are selectable as valid CATEGORY values (S: System Event; P: Patient Related; D: Definition Change; O: Other Event). In the broker hook code, this value will be stored (and can be referenced or changed) in the variable "CAT."

ACTION (#.03)

This field is required for patient-related CATEGORY values. The ACTION field should be set to one of the following values which best describe the function of the RPC. The following codes are selectable as valid ACTION values (A: Additions; D: Deletions; Q: Queries; P: Print; E: Changes; C: Copy; AC: Access to patient information; EA: Emergency access to patient information; U: Change to user privilege; AL: Change to audit log status; ES: Change to encryption status). In the broker hook code, this value will be stored (and can be referenced or changed) in the variable "ACT."

ENTRY DESCRIPTION EXECUTABLE (#.06)

This field should be MUMPS executable code which sets the variable "X" equal to a description of the RPC function.

Note: Any uses of a caret (^) should be input as a tilde (~). They will be translated back to a caret (^) prior to execution of the code.

In the broker hook code, this value will be stored (and can be referenced or changed) in the variable "DESC."

The following examples show possible values for this field:

- S X="EHR: Updated POV for patient"
- S X="EHR: Displayed problem list for patient"
- S X= S X="iCare: print women's health profile"

- S X="iCare: Opened Panel
" _\$\$PNLNAME~BUSAUTIL(DUZ,\$\$SINPUT~BUSAUTIL(2))

INACTIVE (#.07)

This field should be set to 1 if an existing entry should be marked as inactive and should no longer be logged.

DFN LOCATION (#1.01)

If this RPC is patient related and the DFN value is available in either the RPC input parameters or output results, the location of the DFN should be entered here. The format for this field is as follows:

Format: [1]~[2]

[1] – Enter I for input parameter or R for result (where the DFN value or data used to get the DFN value is located).

[2] – Enter the input parameter number or the result field piece or column name. For result references the column number should be listed for CIA Broker RPC calls. For XWB or BMXNet result references the actual column name should be used.

The following examples show possible values for this field:

- I~3 – This will pull the DFN from input parameter 3.
- R~1 – This CIA Broker reference pulls the DFN out of the first piece of the results.
- R~DFN – This XWB/BMXNet reference pulls the DFN out of the column with a header value of DFN.

DFN EXECUTABLE (#1.02)

This field should be MUMPS executable code which sets the variable X equal to the DFN value. If the DFN LOCATION field is populated, this value will already be populated in the variable X prior to the execution of this field. After the execution of this field in the broker hook code, the value of X will be saved into the variable DFN where it can later be referenced.

Note: Any uses of a caret (^) should be input as a tilde (~). They will be translated back to a caret (^) prior to execution of the code.

The following examples show possible values for this field:

- S X=\$P(X,U)

This code takes the result of the DFN LOCATION field and pulls the first caret (^) piece to use as the value of DFN.

- S X=\$\$GET1~DIQ(9000011,X_",",.02,"I")

In this case the DFN LOCATION field set the variable X to the IEN of the PROBLEM entry. This executable code then utilizes that value to pull the DFN from the PROBLEM file entry.

- S:\$P(X,U,8)="" DESC="EHR: Added problem entry for patient",ACT="A" S X=\$P(X,U,7)

In this example (for the BGOPROB SET RPC) the variable X is set as the input parameter string. Piece 8 will be blank for new problems and will be populated with the PROBLEM IEN if it is an update. This code is utilizing this condition to alter the value of the description value (initially set as “EHR: Updated problem entry for patient”) by changing the value of DESC. It then sets the DFN to be the seventh piece of the input string.

VIEN LOCATION (#2.01)

If this RPC is patient related and the VIEN value is available in either the RPC input parameters or output results, the location of the VIEN should be entered here.

Note: If VIEN is populated using this field or the subsequent VIEN EXECUTABLE field, it is not necessary to populate the DFN LOCATION or DFN EXECUTABLE fields, as the DFN value will be automatically calculated using the VIEN.

The format for this field is as follows:

Format: [1]~[2]

[1] – Enter I for input parameter or R for result (where the VIEN value or data used to get the VIEN value is located).

[2] – Enter the input parameter number or the result field piece or column name. For result references the column number should be listed for CIA Broker RPC calls. For BMXNet and XWB, result references the actual column name should be used.

The following examples show possible values for this field:

- I~3 – This will pull the VIEN from input parameter 3.
- R~1 – This CIA Broker reference pulls the VIEN out of the first piece of the results.
- R~HIDDEN_VISIT_IEN – This BMXNet reference pulls the VIEN out of the column with a header value of HIDDEN_VISIT_IEN.

VIEN EXECUTABLE (#2.02)

This field should be MUMPS executable code, which sets the variable X equal to the VIEN value. If the **VIEN LOCATION** field is populated, this value will already be populated in the variable X prior to the execution of this field. After the execution of this field in the broker hook code, the value of X will be saved into the variable VIEN where it can later be referenced.

Note: Any uses of a caret (^) should be input as a tilde (~). They will be translated back to a caret (^) prior to execution of the code.

The following example show possible values for this field:

- S X=\$P(X,U) – This code takes the result of the **VIEN LOCATION** field and pulls the first caret (^) piece to use as the VIEN value.

MULTIPLE INPUT LIST TYPE (#2.03)

This field, used in conjunction with the **MULTIPLE INPUT LIST DELIMITER** field, is used to handle cases where multiple DFN or VIEN values are passed in the input parameters as a delimited list.

If either the DFN or VIEN is passed in as a list of values, the value of this field should be set to either DFN or VIEN.

MULTIPLE INPUT LIST DELIMITER (#2.04)

This field, used in conjunction with the **MULTIPLE INPUT LIST TYPE** field, is used to handle cases where multiple DFN or VIEN values are passed in the input parameters as a delimited list.

The delimiter used to separate the DFN/VIEN values should be entered in this field. Enter a character, \$C(.) value, or U for ""^"". Sample entries for this field might be: “;”, “|”, U, or \$C(28).

The following example shows how these two fields can be utilized. In this example, the DFN LOCATION is set as I~3, the input piece that contains the delimited list of DFN values. The **MULTIPLE INPUT LIST TYPE** field is set to DFN to reflect that this is a list of DFN values. Finally, the **MULTIPLE INPUT LIST DELIMITER** is set to \$C(28) since this character is used as the list delimiter.

DETAIL ENTRY DESC EXECUTABLE (#3)

This optional field should be populated with MUMPS executable code which sets the variable X equal to a comment to associate with the particular DFN/VIEN value detail entry. This field (normally left blank) could be useful for situations where multiple DFN/VIEN values are provided, and it is desired to have custom information saved for each DFN/VIEN value.

NEW VALUE EXECUTABLE (#4)

This optional field should be populated with MUMPS executable code which sets the variable X equal to the new value resulting from the RPC call. This information is not required for this stage of meaningful use certification so it should be populated only if absolutely necessary.

ORIGINAL VALUE EXECUTABLE (#5)

This optional field should be populated with MUMPS executable code which sets the variable X equal to the original value of the data that was modified with the RPC call. This information is not required for this stage of meaningful use certification so it should be populated only if absolutely necessary.

- Variable: CAT – The category value
- Variable: ACT – The action value
- Variable: DESC – The logging entry description value
- Variable: SKIP – If set to 1 the log entry will be skipped for this call
- Array: BUSADVAL – For patient-related calls, this array should be set in the following format:

```
BUSADVAL(#)=DFN^VIEN^EVENT DESCRIPTION^NEW  
VALUE^ORIGINAL VALUE
```

Where:

- # – Record counter (1,2,3...)
- DFN – (Optional for non-patient-related calls) – Pointer to VA PATIENT file (#2)
- VIEN – (Optional for non-visit-related calls) – Pointer to VISIT file (#9000010)
- EVENT DESCRIPTION – (Optional) – Additional detail to log for this entry
- NEW VALUE – (Optional) – New value after call completion, if applicable
- ORIGINAL VALUE – (Optional) – Original value prior to call execution, if applicable

4.3.6.2 Instructions for Including BUSA AUDIT RPC DEFINITIONS in KIDS Build

The BUSA application does not come prepopulated with BUSA AUDIT RPC DEFINITION entries. It is the responsibility of the developers of each application to determine which RPC calls need to be logged. After setting up the definition entries in the **BUSA AUDIT RPC DEFINITION** file, the next step is to mark the entries to be included in a specified KIDS build. The following instructions describe how to mark the appropriate entries and populate the KIDS build so that only the desired entries get included.

4.3.6.2.1 Step 1

Identify entries to be included in the KIDS build and set up the entries in the BUSA RPC TRANSPORT LIST file (#9002319.07).

```

Select OPTION: 1 ENTER OR EDIT FILE ENTRIES
INPUT TO WHAT FILE: BUSA RPC TRANSPORT LIST//
EDIT WHICH FIELD: ALL//
Select BUSA RPC TRANSPORT LIST BUSA RPC DEFINITION: BQI UPDATE DX CAT
...OK? Yes// (Yes)
  Are you adding 'BQI UPDATE DX CAT' as
    a new BUSA RPC TRANSPORT LIST (the 90TH)? No// Y (Yes)
Select SEND IN BUILD: BQI*2.3*3          ICARE MANAGEMENT SYSTEM          ICARE
MANAGEMENT SYSTEM
  Are you adding 'BQI*2.3*3' as a new SEND IN BUILD (the 1ST for this BUSA
RPC TRANSPORT LIST)? No// Y (Yes)
Select SEND IN BUILD:

Select BUSA RPC TRANSPORT LIST BUSA RPC DEFINITION: BQI UPDATE FAMILY
PLANNING
...OK? Yes// (Yes)

  Are you adding 'BQI UPDATE FAMILY PLANNING' as
    a new BUSA RPC TRANSPORT LIST (the 91ST)? No// Y (Yes)
Select SEND IN BUILD: BQI*2.3*3          ICARE MANAGEMENT SYSTEM          ICARE
MANAGEMENT SYSTEM
  Are you adding 'BQI*2.3*3' as a new SEND IN BUILD (the 1ST for this BUSA
RPC TRANSPORT LIST)? No// Y (Yes)
Select SEND IN BUILD:

Select BUSA RPC TRANSPORT LIST BUSA RPC DEFINITION:

```

Figure 4-1: Instructions for Step 1

4.3.6.2.2 Step 2

Include the **BUSA AUDIT RPC DEFINITIONS** file in the application KIDS build with the following parameters:

Include the **BUSA AUDIT RPC DEFINITIONS** file in the KIDS build:

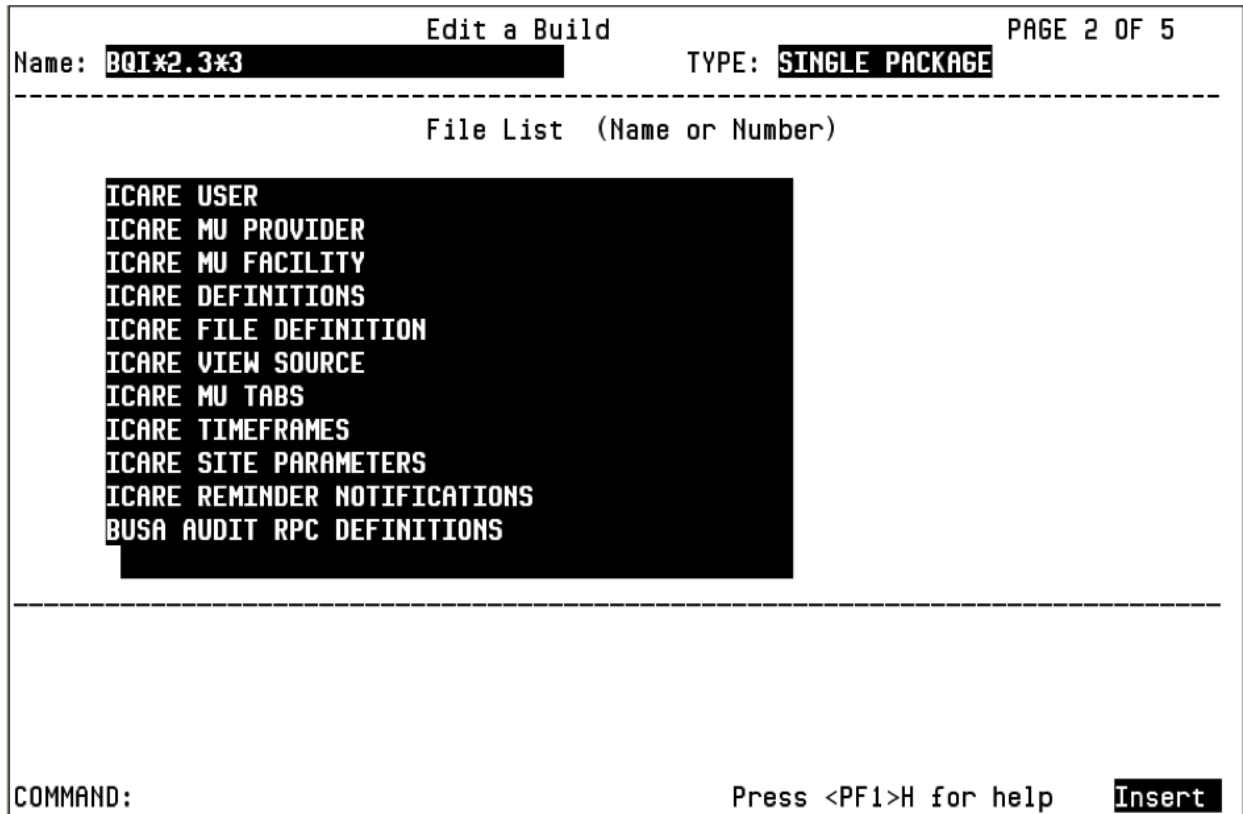


Figure 4-2: Include the BUSA AUDIT RPC DEFINITIONS file in the KIDS build

Set up the file with the following KIDS transport definitions:

```

                                Edit a Build                                PAGE 2 OF 5
Name: BQI*2.3*3                    TYPE: SINGLE PACKAGE
-----
                                File List (Name or Number)
                                DD Export Options
-----
                                File: BUSA AUDIT RPC DEFINITIONS
                                Send Full or Partial DD...: FULL
                                Update the Data Dictionary: NO      Send Security Code: NO
                                Screen to Determine DD Update
                                Data Comes With File...: YES
-----
COMMAND:                               Press <PF1>H for help  Insert

```

Figure 4-3: Set up the file with the KIDS transport definitions

In the **Data Comes With File** page, set up the following properties:

Edit a Build		PAGE 2 OF 5
Name: BQI*2.3*3	TYPE: SINGLE PACKAGE	

File List (Name or Number)		
DD Export Options		
Data Export Options		
Site's Data: REPLACE		
Resolve Pointers: NO	May User Override Data Update: NO	
Data List:		
Screen to Select Data		
I \$\$\$SEND^BUSUTIL(\$G(Y),\$G(XPDA))		

COMMAND:	Press <PF1>H for help	Insert

Figure 4-4: Set the properties on the DO Export Options page

Make sure to enter the following line of code in the **Screen to Select Data** field:

I \$\$\$SEND^BUSUTIL(\$G(Y),\$G(XPDA))

4.3.6.3 Using the ^BUSATRANS Utility

The IHS SECURITY AUDITING package comes with a utility that makes the definition of the RPC entries more convenient. From the command line type the following to initiate the utility:

4.3.6.3.1 Add a New Entry

```
>D ^BUSATRAN

Select REMOTE PROCEDURE NAME:   BJPN SET AS POV
Add entry to transport list? Yes//  YES
CATEGORY: P Patient Related
ACTION: E Changes
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Set prenatal problem as POV"
INACTIVE:
DFN LOCATION:
DFN EXECUTABLE: I~1
VIEN LOCATION:
VIEN EXECUTABLE:
MULTIPLE INPUT LIST TYPE:
MULTIPLE INPUT LIST DELIMITER:
```

```

DETAIL ENTRY DESC EXECUTABLE:
NEW VALUE EXECUTABLE:
ORIGINAL VALUE EXECUTABLE:
ADVANCED DEFINITION EXECUTABLE:
Select SEND IN BUILD: BJPN  PRENATAL CARE MODULE      BJPN  PRENATAL CARE
MODULE
1.0      PRENATAL CARE MODULE      PRENATAL CARE MODULE
Are you adding 'PRENATAL CARE MODULE 1.0' as
a new SEND IN BUILD (the 1ST for this BUSA RPC TRANSPORT LIST)? No// Y
(Yes)
    
```

Figure 4-5: Add a new entry

4.3.6.3.2 Edit an Existing Entry

```

>D ^BUSATRAN

Select REMOTE PROCEDURE NAME:      BJPN SET AS POV

      Select one of the following:

          E          Edit Transport Entry
          D          Delete Transport Entry

Choose operation to perform: E// Edit Transport Entry
CATEGORY: Patient Related//
ACTION: Changes//
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Set prenatal problem as POV"
      Replace

INACTIVE:
DFN LOCATION:
DFN EXECUTABLE: I~1//
VIEN LOCATION:
VIEN EXECUTABLE:
MULTIPLE INPUT LIST TYPE:
MULTIPLE INPUT LIST DELIMITER:
DETAIL ENTRY DESC EXECUTABLE:
NEW VALUE EXECUTABLE:
ORIGINAL VALUE EXECUTABLE:
ADVANCED DEFINITION EXECUTABLE:
    
```

Figure 4-6: Edit an existing entry

4.3.6.3.3 Delete an Existing Entry

```

>D ^BUSATRAN

Select REMOTE PROCEDURE NAME:      BJPN SET AS POV

      Select one of the following:

          E          Edit Transport Entry
          D          Delete Transport Entry

Choose operation to perform: E// Delete Transport Entry
    
```

```
Are you sure you wish to delete the entry? No// YES
TRANSPORT ENTRY DELETED...
Do you wish to delete the BUSA RPC definition entry as well? No// YES
BUSA RPC DEFINITION DELETED...
```

Figure 4-7: Delete an existing entry

5.0 Files and Tables

5.1 File List

Table 5-1 contains a list of new files.

Table 5-1: New files by number, filename, and description

File #	Filename	Description
9002319.01	BUSA AUDIT LOG SUMMARY	This file contains summary (overview) records of logged events.
9002319.02	BUSA AUDIT LOG DETAIL	This file contains detail-level records of logged events. One or more detail records can point to a single summary record.
9002319.03	BUSA AUDIT RPC DEFINITIONS	This file contains a list of RPCs (and their BUSA definitions), which have been set up to be audited in BUSA.
9002319.04	BUSA SWITCH SETTINGS	This file tracks the status of the various auditing on/off switches.
9002319.05	BUSA CACHE CLASS TRANSPORT	This file contains the encrypted BUSA class definitions.
9002319.07	BUSA RPC TRANSPORT LIST	This file is used to control which BUSA AUDIT RPC DEFINITIONS entries should be included in specified application KIDS builds.
9002319.08	BUSA FILEMAN AUDIT INCLUSIONS FILE	This file contains a list of FileMan files that should be audited in BUSA (when FileMan auditing is turned on) because they contain patient related information.
9002319.09	BUSA FILEMAN LOCAL AUDIT DEF	This file contains a snapshot of the patient related FileMan files before they were modified by a change to the BUSA FileMan audit switch. The information is used to revert the FileMan field audit settings back to their prior value if the status of the switch changes.
9002319.11	BUSA AUDIT LOG SUMMARY ARCHIVE	This file contains archived summary records that have been reloaded onto the system for review.
9002319.12	BUSA AUDIT LOG DETAIL ARCHIVE	This file contains archived detail records that have been reloaded onto the system for review.
9002319.13	BUSA ARCHIVE HISTORY	This file tracks the history of archive activities

5.2 File Access

Table 5-2 contains the FileMan access to new files.

Table 5-2: FileMan access by file number and filename

File #	Filename	GL	RD	WR	LYG	DD	DEL
9002319.01	BUSA AUDIT LOG SUMMARY	^BUSAS(@	@	@	@	@
9002319.02	BUSA AUDIT LOG DETAIL	^BUSAD(@	@	@	@	@
9002319.03	BUSA AUDIT RPC DEFINITIONS	^BUSA(9002319.03,	@	@	@	@	@
9002319.04	BUSA SWITCH SETTINGS	^BUSA(9002319.04,	@	@	@	@	@
9002319.05	BUSA CACHE CLASS TRANSPORT	^BUSACLS(@	@	@	@	@
9002319.07	BUSA RPC TRANSPORT LIST	^BUSATR(@	@	@	@	@
9002319.08	BUSA FILEMAN AUDIT INCLUSIONS FILE	^BUSAFMAN(@	@	@	@	@
9002319.09	BUSA FILEMAN LOCAL AUDIT DEF	^BUSAFDEF(@	@	@	@	@
9002319.11	BUSA AUDIT LOG SUMMARY ARCHIVE	^BUSASA(@	@	@	@	@
9002319.12	BUSA AUDIT LOG DETAIL ARCHIVE	^BUSADA(@	@	@	@	@
9002319.13	BUSA ARCHIVE HISTORY	^BUSAAH(@	@	@	@	@

5.3 Cross References

```

9002319.01 (BUSA AUDIT LOG SUMMARY)
  B Regular type cross reference
    LOG TIMESTAMP (#.01)

  C Regular type cross reference
    USER (#.02)

9002319.02 (BUSA AUDIT LOG DETAIL)
  B Regular type cross reference
    SUMMARY ENTRY (#.01)

  C Regular type cross reference
    DFN (#.02)

9002319.03 (BUSA AUDIT RPC DEFINITIONS)
  B Regular type cross reference
    RPC (#.01)

9002319.04 (BUSA SWITCH SETTINGS)
  B Regular type cross reference
    SWITCH TYPE (#.01)
    
```

```

C New style cross reference
SWITCH TYPE (#.01)
SETTING DATE (#.03)

S Regular mumps style cross reference
SWITCH TYPE (#.01)
LOGGING STATUS (#.02)
*This cross reference will only contain the latest status
defined for each switch type

9002319.05 (BUSA CACHE CLASS TRANSPORT)
B Regular type cross reference
PACKAGE NAME (#.01)
B Regular type cross reference
CLASS (Multiple 9002319.061, field #.01)

9002319.07 (BUSA RPC TRANSPORT LIST)
B Regular type cross reference
BUSA RPC DEFINITION (#.01)
B Regular type cross reference
SEND IN BUILD (Multiple 9002319.071, field #.01)

9002319.08 (BUSA FILEMAN AUDIT INCLUSIONS FILE)
B Regular type cross reference
FILE (#.01)
F Regular mumps style cross reference
INCLUDED FILE (#.01)
AUDIT FILE (#.02)

9002319.09 (BUSA FILEMAN LOCAL AUDIT DEF FILE)
B Regular type cross reference
CREATE DATE (#.01)
F Regular mumps style cross reference
FILE NUMBER (Multiple 9002319.91, field #.01)
FIELD NUMBER (Multiple 9002319.911, field #.01)

9002319.11 (BUSA AUDIT LOG SUMMARY ARCHIVE FILE)
B Regular type cross reference
TIMESTAMP (#.01)
C Regular type cross reference
USER (#.02)

9002319.12 (BUSA AUDIT LOG DETAIL ARCHIVE FILE)
B Regular type cross reference
SUMMARY ENTRY (#.01)
C Regular type cross reference
DFN (#.02)

9002319.13 (BUSA ARCHIVE HISTORY FILE)
B Regular type cross reference
ARCHIVE TIMESTAMP (#.01)
C Regular type cross reference
STATUS (#.02)

```

Figure 5-1: Cross references

5.4 Table File

File: 9002319.01 BUSA AUDIT LOG SUMMARY

Global: ^BUSAS(

Field #	Field Name	Subscript	Piece	Type
.01	LOG TIMESTAMP	D0,0	1	D
.02	USER	"	2	P (#200)
.03	CATEGORY	"	3	S (S:System Event, P:Patient Related, D:Definition Event, O:Other Event)
.04	CALL TYPE	"	4	S (R:RPC Call, W:Web Service Call, A:API Call, O:Other)
.05	ACTION	"	5	S (A:Additions, D:Deletions, Q:Queries, P:Print, E:Changes, C:Copy, AC:Access to patient information, EA:Emergency access to patient information, U:Change to user privilege, AL:Change to audit log status, ES:Change to encryption status)
.06	ORIGINATING CALL	"	6	F
1	ENTRY DESCRIPTION	D0,1	1	F
2	HASH	D0,2	1	F

File: 9002319.02 BUSA AUDIT LOG DETAIL

Global: ^BUSAD(

Field #	Field Name	Subscript	Piece	Type
.01	SUMMARY ENTRY	D0,0	1	P (#9002319.01)
.02	DFN	"	2	P (#2)
.03	VISIT	"	3	P (#9000010)
.04	DETAIL EVENT DESCRIPTION	"	4	F
1	NEW VALUE	D0,1	1	W
2	ORIGINAL VALUE	D0,2	1	W
3	HASH	D0,3	1	F

File: 9002319.03 BUSA AUDIT RPC DEFINITIONS**Global: ^BUSA(9002319.03,**

Field #	Field Name	Subscript	Piece	Type
.01	RPC	D0,0	1	P (#8994)
.02	CATEGORY	"	2	S (S:System Event, P:Patient Related, D:Definition Event, O:Other Event)
.03	ACTION	"	3	S (A:Additions, D:Deletions, Q:Queries, P:Print, E:Changes, C:Copy, AC:Access to patient information, U: Change to user privilege, AL:Change to audit log status, ES:Change to encryption status)
.06	ENTRY DESCRIPTION EXECUTABLE	"	6	F
.07	INACTIVE	"	7	S (1:INACTIVE)
1.01	DFN LOCATION	D0,1	1	F
1.02	DFN EXECUTABLE	"	2	F
2.01	VIEN LOCATION	D0,2	1	F
2.02	VIEN EXECUTABLE	"	2	F
2.03	MULTIPLE INPUT LIST TYPE	"	3	S (D:DFN, V:VIEN)
2.04	MULTIPLE INPUT LIST DELIMITER	"	4	F
3	DETAIL ENTRY DESC EXECUTABLE	D0,3	1	F
4	NEW VALUE EXECUTABLE	D0,4	1	F
5	ORIGINAL VALUE EXECUTABLE	D0,5	1	F
6	ADVANCED DEFINITION EXECUTABLE	D0,6	1	F

File: 9002319.04 BUSA SWITCH SETTINGS**Global: ^BUSA(9002319.04,**

Field #	Field Name	Subscript	Piece	Type
.01	SWITCH TYPE	D0,0	1	S (M:Master, B:BMXNet, C:CIA Broker, W:XWB Broker, F:FileMan)
.02	LOGGING STATUS	"	2	S (1:On, 0:Disabled)
.03	SETTING DATE	"	3	D

Field #	Field Name	Subscript	Piece	Type
.04	USER	"	4	P (#200)
.05	DISABLE COMMENT	"	5	F

File: 9002319.05 BUSA CACHE CLASS TRANSPORT**Global: ^BUSACLS(**

Field #	Field Name	Subscript	Piece	Type
.01	PACKAGE NAME	D0,0	1	F
.02	*INSTALL WHERE	"	2	S (R:RPMS, E:ENSRPMS, B:BOTH)
.04	*PATH	"	4	F
1.01	*RPMS FILENAME	D0,1	1	F
1.02	RPMS STATUS	"	2	S (R:READY, I:IN PROGRESS, C:COMPLETE, E:ERROR)
1.03	RPMS DATE/TIME INSTALLED	"	3	D
2.01	*ENSEMBLE FILENAME	D0,2	1	F
2.02	*ENSEMBLE STATUS	"	2	S (R:READY, I:IN PROGRESS, C:COMPLETE, E:ERROR)
2.03	*ENSEMBLE DATE/TIME INSTALLED	"	3	D
10	XML	D0,10	1	W
11	CLASS	D0,11	1	W

File: 9002319.07 BUSA RPC TRANSPORT LIST**Global: ^BUSATR(**

Field #	Field Name	Subscript	Piece	Type
.01	BUSA RPC DEFINITION	D0,0	1	P (#9002319.03)
1	SEND IN BUILD	D0,1,D1		
.01	SEND IN BUILD	D0,1,D1,0	1	P (#9.6)

File: 9002319.08 BUSA FILEMAN AUDIT INCLUSIONS FILE**Global: ^BUSAFMAN(**

Field #	Field Name	Subscript	Piece	Type
.01	INCLUDED FILE	D0,0	1	F
.02	AUDIT FILE	"	2	S (0:Do not audit, 1:Audit the file)
.03	TYPE	"	3	S (1:USER RELATED, 0:PATIENT RELATED)

Field #	Field Name	Subscript	Piece	Type
1	DFN LOGIC	D0,1	1	F
2	VIEN LOGIC	D0,2	1	F

File: 9002319.09 BUSA FILEMAN LOCAL AUDIT DEF

Global: ^BUSAFDEF(

Field #	Field Name	Subscript	Piece	Type
.01	CREATION DATE	D0,0	1	D
.02	SWITCH STATUS	"	2	S (1:ENABLED, 0:DISABLED)
.03	USER	"	3	P (#200)
1	FILE NUMBER	D0,1		
.01	FILE NUMBER	D0,1,D1,0	1	F
1	FIELD NUMBER	D0,1,D1,1		
.01	FIELD NUMBER	D0,1,D1,1,D2,0	1	F
.02	ORIGINAL AUDIT VALUE	"	2	F
.03	NEW AUDIT VALUE	"	3	F

File: 9002319.11 BUSA AUDIT LOG SUMMARY ARCHIVE

Global: ^BUSASA(

Field #	Field Name	Subscript	Piece	Type
.01	TIMESTAMP	D0,0	1	D
.02	USER	"	2	P (#200)
.03	CATEGORY	"	3	S (S:System Event, P:Patient Related, D:Definition Event, O:Other Event)
.04	CALL TYPE	"	4	S (R:RPC Call, W:Web Service Call, A:API Call, O:Other)
.05	ACTION	"	5	S (A:Additions, D:Deletions, Q:Queries, P:Print, E:Changes, C:Copy, AC:Access to patient information, EA:Emergency access to patient information, U: Change to user privilege, AL:Change to audit log status, ES:Change to encryption status)
.06	ORIGINATING CALL	"	6	F
1	ENTRY DESCRIPTION	D0,1	1	F
2	HASH	D0,2	1	F

File: 9002319.12 BUSA AUDIT LOG DETAIL ARCHIVE

Global: ^BUSADA(

Field #	Field Name	Subscript	Piece	Type
.01	SUMMARY ENTRY	D0,0	1	P (#9002319.11)
.02	DFN	"	2	P (#2)
.03	VISIT	"	3	P (#9000010)
.04	DETAIL EVENT DESCRIPTION	"	4	F
1	NEW VALUE	D0,1	0	W
2	ORIGINAL VALUE	D0,2	0	W
3	HASH	D0,3	1	F

File: 9002319.13 BUSA ARCHIVE HISTORY

Global: ^BUSAAH(

Field #	Field Name	Subscript	Piece	Type
.01	ARCHIVE TIMESTAMP	D0,0	1	D
.02	ARCHIVE USER	"	2	P (#200)
.03	ARCHIVE START DATE	"	3	D
.04	ARCHIVE END DATE	"	4	D
.05	FIRST ARCHIVE BUSA RECORD	"	5	N
.06	LAST ARCHIVE BUSA RECORD	"	6	N
.07	TOTAL RECORDS IN ARCHIVE	"	7	N
.08	VERIFICATION COMPLETE	"	8	S (1:YES)
.09	VERIFICATION DATE	"	9	D
.1	VERIFICATION USER	"	10	P (#200)
.11	ARCHIVE FILENAME	"	11	F
.12	ARCHIVE PURGE DATE	"	12	D
.13	ARCHIVE PURGE USER	"	13	P (#200)
.14	STATUS	"	14	S (C:CREATED, V:VERIFIED, A:ARCHIVED)
1	RESTORED ARCHIVE	D0,1		
.01	RESTORED ARCHIVE DATE	D0,1,D1,0	1	D
.02	RESTORED ARCHIVE USER	"	2	P (#200)
.03	RESTORED ARCHIVE RECORDS	"	3	N
.04	RESTORED ARCHIVE PURGE DATE	"	4	D
.05	RESTORED ARCHIVE PURGE USER	"	5	P (#200)

6.0 External Relations

6.1 Callable Routines

Table 6-1: Callable routines and descriptions

Routine Called	Description
RPC^BUSARPC	BUSA LOG SECURITY AUDIT ENTRY – RPC call to log BUSA activity

6.2 Published Entry Points

Table 6-2: Published entry points and descriptions

Routine Called	Description
FAUD^BUSAAPI	Return whether a FileMan audit should be performed for a particular file
FMENT^BUSAAPI	Log a BUSA entry for a FileMan AUDIT file addition
LOG^BUSAAPI	Records a BUSA entry based on supplied input criteria
BMX^BUSARPC	BUSA RPC front end for BMXNet broker
CIA^BUSARPC	BUSA RPC front end for CIA broker
XWB^BUSARPC	BUSA RPC front end for XWB broker
SINPUT^BUSAUTIL	Return RPC call input parameter value
SOUTPUT^BUSAUTIL	Return RPC call output field value
VAL^BUSAUTIL	Return information located in the specified location
CHECK^BUSAAPIR	Return whether specified remediation type/record is set up for remediation

6.3 Exported Options

Table 6-3: Exported options and descriptions

Option Name	Description
BUSA ARCHIVE BUSA INFORMATION	Option which allows sites to export audit file data to external archive files
BUSA ARCHIVE MENU	Menu containing the list of BUSA archive options
BUSA ARCHIVE REPORT	Option which provides information on previous audit archive activities
BUSA LOAD BUSA ARCHIVE FILE	Option which allows external archive files to be imported into the BUSA summary and detail archive files so that they can be reviewed

Option Name	Description
BUSA PURGE BUSA RECORDS	Option which allows sites to purge information from the BUSA summary and detail files once they have been archived to external files and verified
BUSA REMOVE RESTORED RECORDS	Option which allows re-loaded archive information to be removed from the system after it is no longer needed
BUSA SECURITY EDIT	Option which allows the security audit logging to be turned on and off
BUSA VERIFY ARCHIVE	Option which analyzes exported archive files to ensure they are properly formatted and uncorrupted
BUSAMENU	Option which allows users to access the archive functionality and manage the configuration for the BUSA application
BUSARPC	Contains list of BUSA callable RPCs

7.0 Internal Relations

All functions within this application work independently.

There are no documented internal relations in BUSA.

8.0 Archiving and Purging

8.1 Purging functionality included with the Patch 4 release

A number of sites have encountered issues with invalid entries in their ADT/HL7 PIVOT file (#391.71) causing their BUSA Cache database file size to grow at an exponential rate. While many sites have worked with MPI application developers to slow down the exponential growth of the database, the increased size of the database was not addressed. BSTS V1.0 patch 4 contains functionality to help address the database size issue. The Patch 4 installation prompts whether a background process should be kicked off to clean up the MPI related BUSA summary entries. The process, if run, will search through all of the entries in the BUSA AUDIT LOG SUMMARY file, looking for entries added because of the invalid entries in the ADT/HL7 PIVOT file. The background process will remove each invalid entry it finds in the file. Once the process completes its search of the file, it will set up a new BUSA Remediation entry for the removed records to ensure that they will not show up as missing BUSA IEN errors in the EPCS incident reports.

Note: Removing the entries will not reduce the size of the BUSA Cache database. In order to achieve this, the database must then be compacted and truncated. For any questions on how to do this, please contact the help desk for assistance.

If the site does not wish to kick off the background process during the BUSA Patch 4 installation, they can always kick off the process later by typing the following command at the programmer's prompt:

```
>D TSKAGMP^BUSA1P04
```

8.2 Archiving and Purging Functionality Included with the Patch 3 Release

Archiving and purging functionality was included in the Patch 3 release to allow sites to export audit events. Sites must keep two years of audit records available on their RPMS system to remain compliant with EPCS and 2015 CHIT certification requirements. However, audit events need to be maintained for the length of time defined by the organization's applicable record retention schedule. Due to the additional audit events being logged with this release and in the previous EPCS release, sites need to offload audit events into an archive to preserve RPMS database space.

The archiving functionality is documented in the associated *IHS User Security Audit (BUSA) User Manual*.

9.0 Documentation Resources

This section describes a few methods to generate BUSA technical documentation.

9.1 System Documentation

Online VPS system documentation can be generated through the use of several Kernel options, including, but not limited to:

- %INDEX
- Menu Management
- Inquire Option
- Print Option File
- VA FileMan
- Data Dictionary Utilities
- List File Attributes

9.2 %INDEX

Running %INDEX for a specified set of routines allows users to discover any deviations from RPMS programming standards that exist in the selected routines and to see how routines interact with one another (i.e., which routines call or are called by other routines).

To run %INDEX for the Patient Registration package, type the **BUSA** namespace at the “Routine(s)?” prompt.

9.3 List File Attributes Option

This VA FileMan option allows users to generate documentation pertaining to files and file structure. Using the standard format of this option yields the following data dictionary information for a specified file:

- File name and description
- Identifiers
- Cross-references
- Files pointed to by the file specified
- Files that point to the file specified
- Input, print, and sort templates

In addition, the following applicable data is supplied for each field in the file:

- Field name, number, title, and description
- Global location
- “Help” prompt
- Cross-references
- Input transform
- Date last edited
- Notes

Using the Global Map format of this option generates an output that lists the following information:

- All cross-references for the file selected
- Global location of each field in the file
- Input, print, and sort templates

10.0 SAC Requirements and Exemptions

No exemptions are noted at this time.

11.0 Templates, Forms, and Protocols

11.1 Print Templates

There are no print templates in BUSA.

11.2 Sort Templates

There are no sort templates in BUSA.

11.3 Input Templates

BUSA UPDATE.

11.4 List Templates

There are no list templates in BUSA.

11.5 Forms

There are no forms in BUSA.

11.6 Protocols

There are no protocols in BUSA

12.0 Sample BUSA Audit RPC Definitions

12.1 BEHOENCV DETAIL

```
RPC: BEHOENCV DETAIL
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Displayed visit or appointment
detail for patient"
VIEN LOCATION: I~2
VIEN EXECUTABLE: S X=$P(X,";",4)
```

12.2 BEHOENCX VISITLST

```
RPC: BEHOENCX VISITLST
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Retrieved list of patient visits"
VIEN LOCATION: R~1
VIEN EXECUTABLE: S X=$P(X,";",4)
```

12.3 BEHOENCV LIST

```
RPC: BEHOENCV LIST
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Displayed a list of
appointments/visits for patient"
DFN LOCATION: I~1
```

12.4 BEHOENCX FETCH

```
RPC: BEHOENCX FETCH
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Selected patient visit"
DFN LOCATION: I~1
VIEN LOCATION: I~2
VIEN EXECUTABLE: S X=$$VFETCH~BUSUTIL(X,.DESC)
```

12.5 BEHOENCX UPDPRV

```
RPC: BEHOENCX UPDPRV
CATEGORY: Patient Related
ACTION: Changes
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Updated visit provider"
VIEN LOCATION: I~2
VIEN EXECUTABLE: S X=$P(X,";",4)
```

12.6 BEHOENCX PTINFO

```
RPC: BEHOPTCX PTINFO
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Selected/Refreshed patient"
DFN LOCATION: I~1
```

12.7 BGOVPOV SET

```
RPC: BGOVPOV SET
CATEGORY: Patient Related
ACTION: Changes
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Set Patient POV for visit"
VIEN LOCATION: I~1
VIEN EXECUTABLE: S X=$P(X,U,2)
```

12.8 BGOVPOV GET

```
RPC: BGOVPOV GET
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Displayed patient purpose of visit
information"
VIEN LOCATION: I~1
```

12.9 BGOPROB SET

```
RPC: BGOPROB SET
CATEGORY: Patient Related
ACTION: Changes
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Updated problem entry for patient"
DFN LOCATION: I~1
DFN EXECUTABLE: S:$P(X,U,8)=" " DESC="EHR: Added problem entry for
patient",ACT
="A" S X=$P(X,U,7)
```

12.10 BQI GET DX CAT BY PANEL

```
RPC: BQI GET DX CAT BY PANEL
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="iCare: Displayed patient diagnostic
tags"
DFN LOCATION: I~3
MULTIPLE INPUT LIST TYPE: DFN
MULTIPLE INPUT LIST DELIMITER: $C(28)
```

12.11 BQI LOOKUP PATIENTS

```
RPC: BQI LOOKUP PATIENTS
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="iCare Patient Lookup on
'_'_$$SINPUT~BUSUTIL(1)_"'"
DFN LOCATION: R~DFN
```

12.12 BJPN UPDATE PROBLEM

```
RPC: BJPN UPDATE PROBLEM
CATEGORY: Patient Related
ACTION: Changes
ENTRY DESCRIPTION EXECUTABLE: S X="EHR: Updated prenatal PIP problem"
VIEN LOCATION: I~1
```

12.13 BJPN UPDATE PROBLEM

```
RPC: BQI GET DFN LIST BY PANEL
CATEGORY: Patient Related
ACTION: Queries
ENTRY DESCRIPTION EXECUTABLE: S X="iCare: Opened Panel
"_'_$$PNLNAME~BUSUTIL(DUZ,$$SINPUT~BUSUTIL(2))
DFN LOCATION: R~DFN
```

13.0 Ensemble Classes Used by BUSA

13.1 BUSA Report Classes

The BUSA application has three Zen reporting utilities available to access BUSA data. As an extra level of security, it also stores a list of approved users of the BUSA reporting utilities. The Ensemble classes in Table 2-1 are being used by the BUSA application.

Table 13-1: Ensemble classes and descriptions

Class	Description
ArchiveReportPage.cls	Primary class for the new Archive Zen reporting tool
ArchiveReportingApp.cls	Application class for the new Archive Zen reporting tool
EPCSMainReportPage.cls	Primary class for the EPCS Zen reporting tool
EPCSReportingApp.cls	Application class for the EPCS Zen reporting tool
MainReportPage.cls	Primary class for the BUSA Zen reporting tool
ReportingApp.cls	Application class for the BUSA Zen reporting tool
Users.cls	Class containing approved users of the Zen reporting tools
Utilities.cls	Class containing various methods utilized by the Zen reporting tools

13.2 BUSA Remediation Classes

The BUSA application contains a remediation utility that enables users to enter information that should be filtered out of reports and other displays. An RPMS API call (\$\$CHECK^BUSAAPIR) has been made available which returns whether a specified type/record combination has been remediated.

Table 13-2: Remediation classes and descriptions

Class	Description
BUSA.Remediation.DataModels.RecordHistory.cls	Class containing the structure of the record history information
BUSA.Remediation.DataModels.Remediation.cls	DataModel class of remediation information used by the remediation utility
BUSA.Remediation.Application.cls	Remediation application class
BUSA.Remediation.Records.cls	Class containing the remediated record information
BUSA.Remediation.Users.cls	Class containing the approved users of the remediation utility

Class	Description
BUSA.RemediationLogin.cls	Primary class for the remediation login screen
BUSA.RemediationUtility.cls	Primary class for the remediation data entry utility

Glossary

Electronic Health Record

An application used by medical organizations to track patient medical records and care.

Meaningful Use

Meaningful Use is a term used by Centers for Medicare and Medicaid Services (CMS) to ensure that providers and hospitals that have adopted certified Electronic Health Record (EHR) are using the technology to further the goals of information exchange among health care professionals. Eligible Providers (EPs) and Eligible Hospitals (EHs) will achieve meaningful use if the EP or EH:

- (a) demonstrates use of certified EHR technology in a meaningful manner,
- (b) demonstrates the certified EHR technology provides for electronic exchange of health information to improve quality of care, and
- (c) uses certified EHR technology to submit information on clinical quality and other measures.

Resource and Patient Management System

A series of integrated software components that includes clinical, administrative, and financial functions.

Acronym List

Acronym	Meaning
API	Application Programming Interface
CHIT	Certified Health Information Technology
CMS	Centers for Medicare and Medicaid Services
EH	Eligible Hospital
EHR	Electronic Health Record
EP	Eligible Provider
EPCS	Electronic Prescribing of Controlled Substances
IHS	Indian Health Service
OIT	Office of Information and Technology
RPC	Remote Procedure Call
RPMS	Resource and Patient Management System
SAC	Standards and Conventions

Contact Information

If you have any questions or comments regarding this distribution, please contact the IHS IT Service Desk.

Phone: (888) 830-7280 (toll free)

Web: <https://www.ihs.gov/itsupport/>

Email: itsupport@ihs.gov