RESOURCE AND PATIENT MANAGEMENT SYSTEM

IHS USER SECURITY AUDIT Application Programming Interface

(BUSA)

Technical Manual

Version 1.0
August 2019

Office of Information Technology
Division of Information Resource Management

Table of Contents

Document Revision History

Version	Date	Author	Description	Sections
1.0	8/2019	GDIT SESS team	Initial 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 contains 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 has been provided to allow reporting on specific information from Electronic Prescribing of Controlled Substances (ECPS).

1.0 Introduction

The BUSA package is a component of the Indian Health Service (IHS) Resource and Patient Management System (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.

The following table shows the prerequisite patch requirements.

Package and Version	Associated Patch Designation(s)	Brief Patch Description
None	None	None

2.1 System Requirements

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

Module	Minimum Version	Recommended Version
BMW	2013-05-20 or greater	2016-04-26 (V2016.2)
VA FileMan (DI)	v22.0 Patch 1017	
IHS/VA Utilities (XB)	v3.0 through Patch 11	
IHS Kernel Toolkit (XT)	V7.3 through Patch 1017	
VA Kernel (XU)	v8.0 Patch 1017	V8.0 Patch 1019

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.

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.

3.0 Menu Diagram

RPMS menus in the BUSA system:

Edit Security Audit [BUSA SECURITY EDIT]

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

Option requires the BUSAZMGR security key.

4.0 Routine Descriptions

4.1 Routine List

This routine list includes all BUSA routines:

Routine	Routine	Routine
BUSA1P01	BUSA1PRE	BUSAACVR
BUSAAPI	BUSABQI	BUSACLAS
BUSAMAG1	BUSAOPT	BUSARPC
BUSATRAN	BUSAUTIL	

4.2 Routines with Description

This routine list describes each routine in this version.

Routine	Description
BUSA1P01	Patch 1 Post-installation routine
BUSA1PRE	Pre/Post-installation program
BUSAACVR	User access checks for the BUSA Audit Report Utility
BUSAAPI	Main API front end routine
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
BUSATRAN	BUSA RPC definition routine
BUSAUTIL	Utility function calls

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. The following table describes the input parameters expected by this API call.

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’

Parameter	Data Type	Description
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)
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.”
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 ‘1’ if a HASH value should be stored for the resulting BUSA summary 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 gets 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 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.

4.3.3.1 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). 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

4.3.3.2 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.

4.3.3.2.1 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)

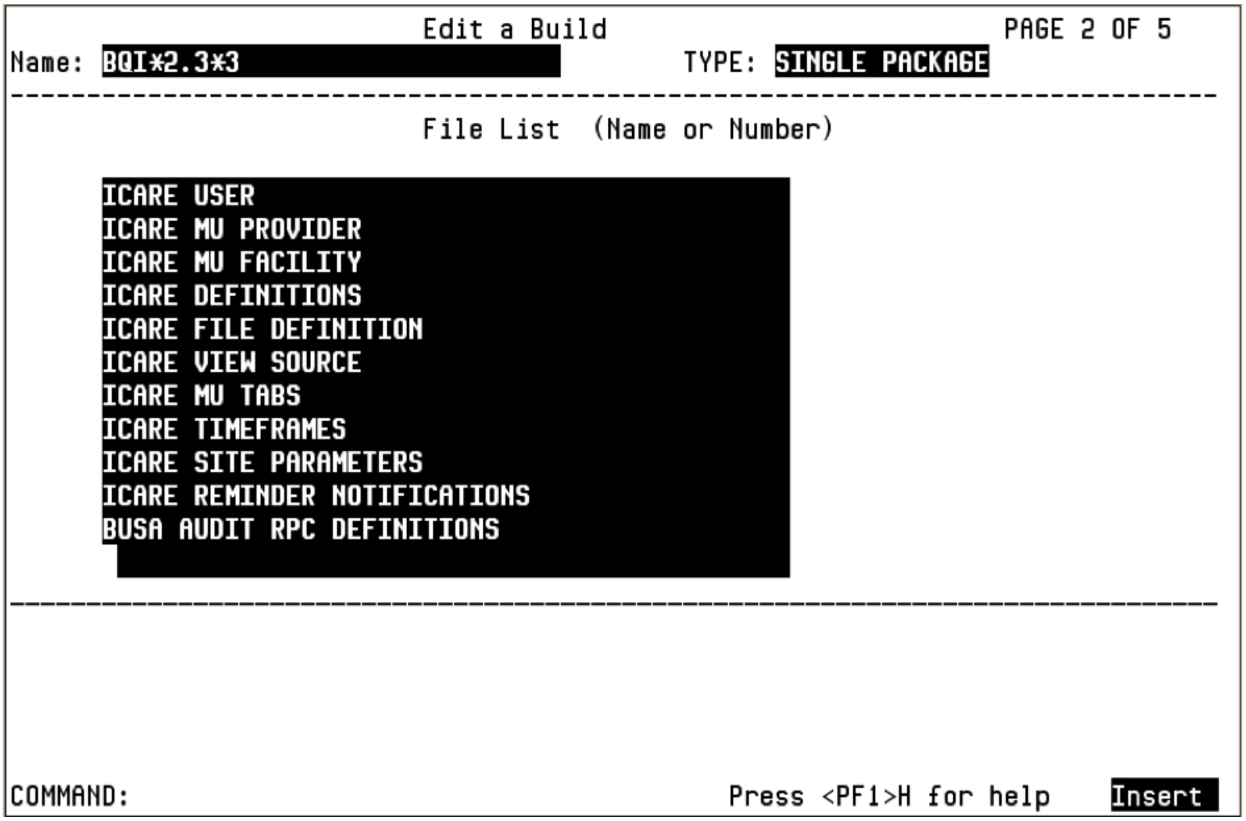
```
Select OPTION: 1  ENTER OR EDIT FILE ENTRIES
INPUT TO WHAT FILE: BUSA RPC TRANSPORT LIST//
EDIT WHICH FIELD: ALL//
Select BUSA RPC TRANSPORT LIST BUSA RPC DEFINITION: BQI UPDATE DX CAT
         ...OK? Yes//   (Yes)
  Are you adding 'BQI UPDATE DX CAT' as
    a new BUSA RPC TRANSPORT LIST (the 90TH)? 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 T
RANSPORT 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
MANAGEME
NT SYSTEM
  Are you adding 'BQI*2.3*3' as a new SEND IN BUILD (the 1ST for this BUSA
RPC T
RANSPORT LIST)? No// Y  (Yes)
Select SEND IN BUILD:

Select BUSA RPC TRANSPORT LIST BUSA RPC DEFINITION:
```

Figure 4-1: Instructions for Step 1

4.3.3.2.2 Step 2

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

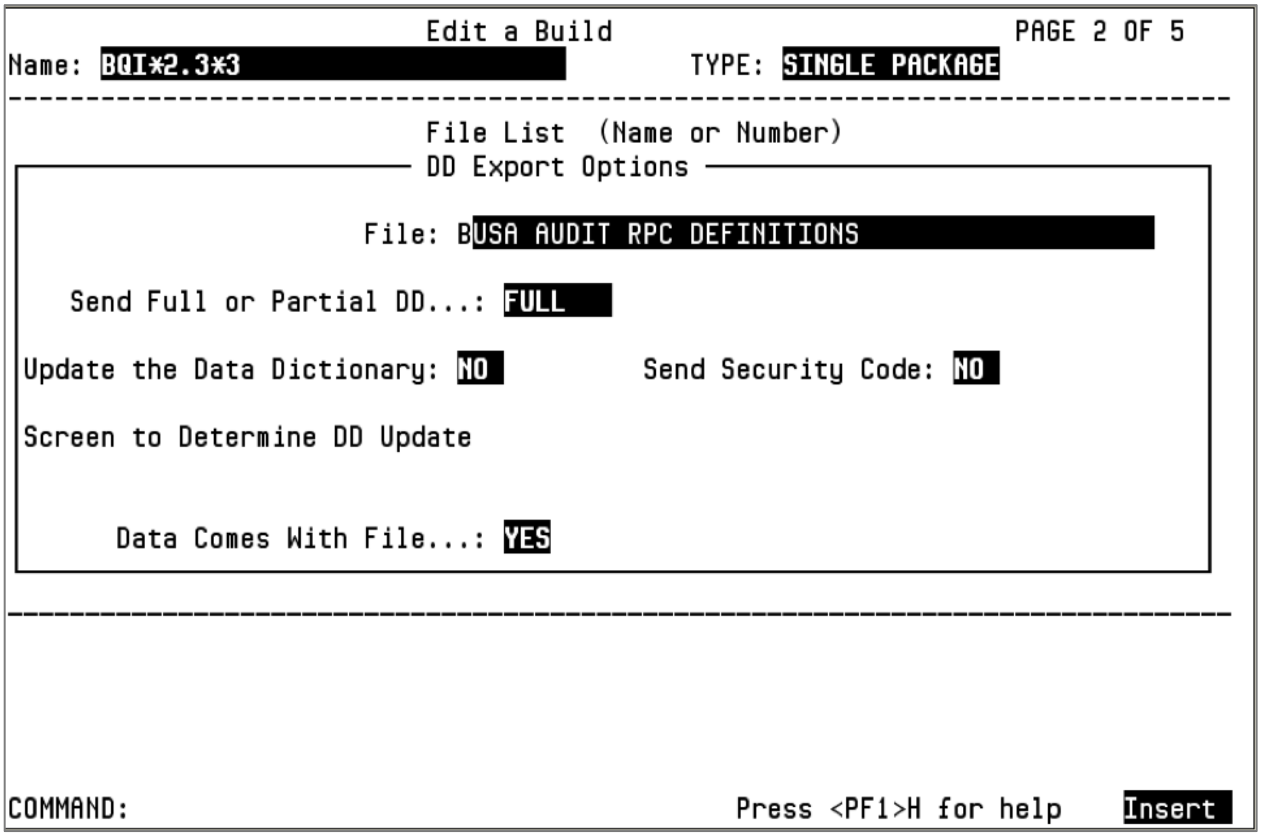
Include the BUSA AUDIT RPC DEFINITIONS file in the KIDS build:



```
                              Edit a Build                          PAGE 2 OF 5
Name: BQI*2.3*3                                  TYPE: SINGLE PACKAGE
-------------------------------------------------------------------------------
                              File List  (Name or Number)

    ICARE USER
    ICARE MU PROVIDER
    ICARE MU FACILITY
    ICARE DEFINITIONS
    ICARE FILE DEFINITION
    ICARE VIEW SOURCE
    ICARE MU TABS
    ICARE TIMEFRAMES
    ICARE SITE PARAMETERS
    ICARE REMINDER NOTIFICATIONS
    BUSA AUDIT RPC DEFINITIONS

-------------------------------------------------------------------------------



COMMAND:                                            Press <PF1>H for help    Insert
```

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

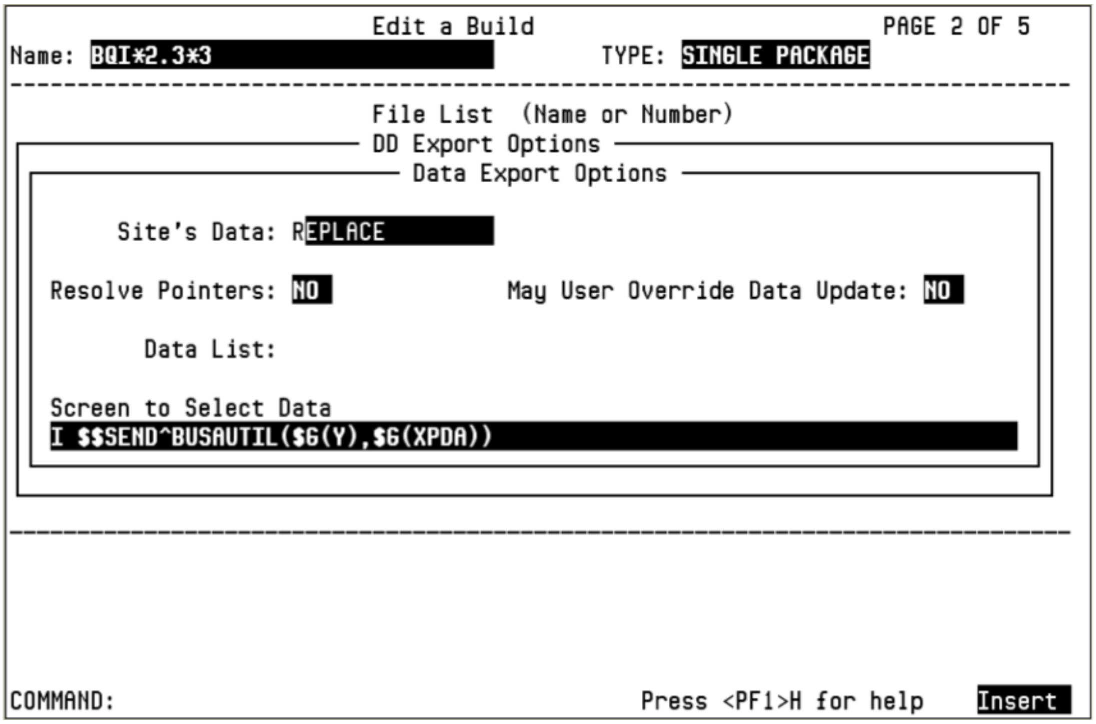
Set up the file with the following KIDS transport definitions:



```
                              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^BUSAUTIL($G(Y),$G(XPDA))

-------------------------------------------------------------------------------



COMMAND:                                         Press <PF1>H for help    Insert
```

Figure 4-4: Set up the properties on this page

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

I $$SEND^BUSAUTIL($G(Y),$G(XPDA))

4.3.3.3 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:

4.3.3.3.1 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

4.3.3.3.2 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

4.3.3.3.3 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

The following table contains a list of new files.

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.

5.2 File Access

The following table contains the FileMan access to new files.

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(	@	@	@	@	@

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)

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 BUILDING (Multiple 9003219.071, field #.01)
```

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
.04	CALL TYPE	”	4	S

Field #	Field Name	Subscript	Piece	Type
.05	ACTION	”	5	S
.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

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
.03	ACTION	"	3	S
.06	ENTRY DESCRIPTION EXECUTABLE	"	6	F
.07	INACTIVE	"	7	S
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
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
.02	LOGGING STATUS	“	2	S
.03	SETTING DATE	"	3	D
.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
.04	*PATH	"	4	F
1.01	*RPMS FILENAME	D0,1	1	F
1.02	RPMS STATUS	“	2	S
1.03	RPMS DATE/TIME INSTALLED	“	3	D
2.01	*ENSEMBLE FILENAME	D0,2	1	F
2.02	*ENSEMBLE STATUS	“	2	S
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)

5.5 Callable Routines

Name	Tag	Routine
BUSA LOG SECURITY AUDIT ENTRY	RPC	BUSARPC

5.6 Published Entry Points

```
BUSAAPI.INT
LOG(TYPE,CAT,ACTION,CALL,DESC,DETAIL) ;PEP - Log Security Audit Entries

BUSARPC.INT
BMX(BUSAP)  ;PEP - Log audit entry from BMXNet
CIA(XWBPTYPE,RTN,BUSAARY) ;PEP - Log audit entry from CIA BrokerXWB(BUSAP) ;PEP -
Log audit entry from XWB Broker

BUSAUTIL.INT
SINPUT(PARM) ;PEP - Return Single Input Parameter Value
SOUTPUT(PARM) ;PEP - Return Single Output Parameter Value
VAL(TYPE,LOCATION,BROKER,RPC,RVAL) ;PEP - Return information located in specified
location
```

Figure 5-2: Published entry points

6.0 External Relations

6.1 External Calls

6.2 Callable Routines – Published Entry Points

Routine Called	Description
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

6.3 Exported Options

Option Name	Description
BUSA SECURITY EDIT	Edit Security Audit (BUSA SECURITY EDIT)
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

There is no archiving or purging with this release of BUSA.

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 AUDIT INPUT

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~BUSAUTIL(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~BUSAUTIL(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
"_$$PNLNAME~BUSAUTIL(DUZ,$$SINPUT~BUSAUTIL(2))
DFN LOCATION: R~DFN
```

13.0 Ensemble Classes Used by BUSA Reports

The BUSA application has two 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 following Ensemble classes are being used by the BUSA application.

Class	Description
EPCSMainReportPage.cls	Primary class for the new EPCS Zen reporting tool
EPCSReportingApp.cls	Application class for the new 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

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 Programmer Interface
BUSA	Namespace for BUSA files and routines
CMS	Centers for Medicare and Medicaid Services
EHR	Electronic Health Record
EHs	Eligible Hospitals
EPCS	Electronic Prescribing of Controlled Substances
EPs	Eligible Providers
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 OIT Help Desk (IHS).

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

Web: https://www.ihs.gov/helpdesk/

Email: support@ihs.gov