BROKER SECURITY ENHANCEMENT (BSE)

SUPPLEMENT TO PATCH DESCRIPTION

RPC Broker Patch XWB*1.1*45 &
Kernel Patch XU*8.0*404

August 2006

Department of Veterans Affairs
VistA Health Systems Design & Development (HSD&D)
Infrastructure & Security Services (ISS)

Revision History

Documentation Revisions

The following table displays the revision history for this manual. Revisions to the documentation are based on patches and new versions released to the field.

Date	Revision	Description	Author(s)
08/10/06	1.0	Initial Broker Security Enhancement (BSE) software and documentation release. **Software Patches:** • **XU*8.0*404** • **XWB*1.1*45**	ISS BSE Development Team Oakland, CA and Bay Pines, FL Office of Information Field Office (OIFO) • Project Manager—Jean Sheppard • Project Planner—Laura Rowland • Developers—Wally Fort & Joel Ivey • SQA—Matt Alderman • Functional Analyst—Lauren Gorgoglione • Technical Writer—Thom Blom • Security Engineer—Jack Michalek • Release Mangers—Mike Palmer and Lewis Tillis

Table i. Documentation revision history

Patch Revisions

For the current patch history related to this software, please refer to the Patch Module on FORUM.

NOTE: The Broker Security Enhancement (BSE)-related software comprises two patches and software releases from the following VistA applications (listed alphabetically):

- Kernel—Kernel Patch XU*8.0*404
- RPC Broker—RPC Broker Patch XWB*1.1*45

REF: For the specific VistA M Server software patches required for the implementation of BSE, please refer to Table 3-2 in Chapter 3, "BSE Installation Instructions for Developers" in this manual.

Contents

Figures and Tables

Acknowledgements

The Broker Security Enhancement (BSE) Project Team consists of the following Development and Infrastructure Services (DaIS) and Infrastructure & Security Services (ISS) personnel (listed alphabetically within each category/title):

- ISS Program Manager—Larry Weldon
- ISS Project Manager—Jean Sheppard
- Centralized Planner Support Team (CPST)—Laura Rowland
- Developers—Wally Fort and Joel Ivey
- Functional Analyst—Lauren Gorgoglione
- Software Quality Assurance (SQA)—Matt Alderman
- Technical Writer—Thom Blom
- Security Engineer—Jack Michalek
- Release Mangers—Mike Palmer and Lewis Tillis

The BSE Project Team would like to thank the following sites/organizations/personnel for their consultation and assistance in reviewing and/or testing the BSE-related software and documentation (listed alphabetically):

- Compensation And Pension Records Interchange (CAPRI) Development Team
- Chief Business Office (CBO) Effort for Centralizing Fiscal Process Development Team (Jerry Sicard Requestor; NSR # 20050614)
- VistAWeb Development Team

Orientation

This *Supplement to Patch Description Manual* is intended for use in conjunction with the Broker Security Enhancement (BSE) Project. It outlines the details of the BSE-related software and gives guidelines on how the software is used within Veterans Health Information Systems and Technology Architecture (VistA).

The intended audience of this manual is all primary (key) stakeholders. The primary stakeholders include:

- VistA Infrastructure and Security Services (ISS) Development Team.
- VistA application services, organizations, and developers that create, maintain, and run RPC Broker Delphi-based applications under Health Systems Design & Development (HSD&D). For example:
  - Hank Rappaport, MD (Requestor)
  - Blind Rehab (BR)
  - Compensation And Pension Records Interchange (CAPRI)
  - Cardiac Assessment Reporting and Tracking System - Cath Laboratory (CART-CL)
  - Chief Business Office (CBO) Effort for Centralizing Fiscal Process (Jerry Sicard Requestor)
  - Common Services (CS)
  - Computerized Patient Record System (CPRS) V. 26
  - Computerized Patient Record System Re-engineering (CPRS-R)
  - CPRS-Remote Data Views (RDV)
  - Health Revenue Center (HRC)
  - Health Systems Implementation Training and Enterprise Support (HSITES) Release Managers
  - Move!
  - Remote VistA Imaging
  - VistALink
  - VistAWeb
- Information Resource Management (IRM) and Information Security Officers (ISOs) at Veterans Affairs Medical Centers (VAMCs) responsible for computer management and system security.
- Enterprise VistA Support (EVS).
- All other VAMC personnel who will be using RPC Broker Delphi-based and BSE-enabled VistA Graphical User Interface (GUI) applications.

How to Use this Manual

This manual is divided into three major parts:

- User Guide—Provides general overview of the BSE Project.
- Developer Guide—Provides step-by-step instructions for VistA developers to follow and Application Program Interfaces (APIs) to use when writing RPC Broker Delphi-based and BSE-enabled applications.
- Systems Management Guide—Provides implementation, maintenance, and security overview for IRM and ISO personnel.

Throughout this manual, advice and instructions are offered regarding the use of the BSE-related software and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VistA) software products.

Official permission to use the BSE functionality is limited. Permission to use the BSE functionality is granted on a case-by-case basis. Applications will be required to send out a patch requesting an entry be made in the REMOTE APPLICATION file (#8994.5).

This manual uses several methods to highlight different aspects of the material:

- Various symbols are used throughout the documentation to alert the reader to special information. The following table gives a description of each of these symbols:

Symbol	Description
	**NOTE/REF:** Used to inform the reader of general information including references to additional reading material.
	**CAUTION or DISCLAIMER:** Used to inform the reader to take special notice of critical information.

Table ii. Documentation symbol/term descriptions

- Descriptive text is presented in a proportional font (as represented by this font).
- Sample HL7 messages, "snapshots" of computer online displays (i.e., roll-and-scroll screen or character-based screen captures/dialogues), and computer source code, if any, are shown in a *non*-proportional font and enclosed within a box.
  - User's responses to online prompts and some software code reserved/key words will be boldface.
  - References to "**<Enter>**" within these snapshots indicate that the user should press the **Enter** key on the keyboard. Other special keys are represented within < > angle brackets. For example, pressing the **PF1** key can be represented as pressing **<PF1>**.
  - Author's comments, if any, are displayed in italics or as "callout" boxes.

    

    NOTE: Callout boxes refer to labels or descriptions usually enclosed within a box, which point to specific areas of a displayed image.

- Delphi/Object Pascal software code, variables, and file/folder names can be written in lower or mixed case. All Object Pascal reserved words are boldface.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names, and security keys (e.g., the XUPROGMODE key).

How to Obtain Technical Information Online

Exported VistA M Server-based file, routine, and global documentation can be generated through the use of Kernel, MailMan, and VA FileMan utilities.

NOTE: Methods of obtaining specific technical information online will be indicated where applicable under the appropriate topic.

Help at Prompts

VistA M Server-based software provides online help and commonly used system default prompts. Users are encouraged to enter question marks at any response prompt. At the end of the help display, you are immediately returned to the point from which you started. This is an easy way to learn about any aspect of VistA M Server-based software.

Obtaining Data Dictionary Listings

Technical information about VistA M Server-based files and the fields in files is stored in data dictionaries (DD). You can use the List File Attributes option on the Data Dictionary Utilities submenu in VA FileMan to print formatted data dictionaries.

REF: For details about obtaining data dictionaries and about the formats available, please refer to the "List File Attributes" chapter in the "File Management" section of the *VA FileMan Advanced User Manual*.

Assumptions About the Reader

This manual is written with the assumption that the reader is familiar with the following:

- VistA computing environment:
  - Kernel—VistA M Server software
  - Remote Procedure Call (RPC) Broker—VistA Client/Server software
  - VA FileMan data structures and terminology—VistA M Server software
- Microsoft Windows environment
- M programming language
- Object Pascal programming language/Borland Delphi Integrated Development Environment (IDE)—RPC Broker

This manual provides an overall explanation of configuring BSE and the functionality contained in the BSE-related software (i.e., Kernel Patch XU*8.0*404 & RPC Broker Patch XWB*1.1*45). However, no attempt is made to explain how the overall VistA programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA home pages on the World Wide Web (WWW) and VA Intranet for a general orientation to VistA. For example, go to the Veterans Health Administration (VHA) Office of Information (OI) Health Systems Design & Development (HSD&D) Home Page at the following Intranet Web address:

http://vista.med.va.gov/

Reference Materials

Readers who wish to learn more about the BSE-related software should consult the following:

- *Broker Security Enhancement (BSE) Supplement to Patch Description (Kernel Patch XU*8.0*404 & RPC Broker Patch XWB*1.1*45*), this manual
- *RPC Broker Getting Started with the RPC Broker Development Kit (BDK)*
- *RPC Broker Developer's Guide (online help)*
- *RPC Broker Systems Manual*
- *RPC Broker Technical Manual*
- *Kernel Programmer Manual*
- *Kernel Systems Manual*

VistA documentation is made available online in Microsoft Word format and Adobe Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following Web address:

http://www.adobe.com/

REF: For more information on the use of the Adobe Acrobat Reader, please refer to the Adobe Acrobat Quick Guide at the following Web address:

http://vista.med.va.gov/iss/acrobat/index.asp

VistA documentation can be downloaded from the Health Systems Design and Development (HSD&D) VistA Documentation Library (VDL) Web site:

http://www.va.gov/vdl/

VistA documentation and software can also be downloaded from the Enterprise VistA Support (EVS) anonymous directories:

- Albany OIFO ftp.fo-albany.med.va.gov
- Hines OIFO ftp.fo-hines.med.va.gov
- Salt Lake City OIFO ftp.fo-slc.med.va.gov

- Preferred Method download.vista.med.va.gov

  This method transmits the files from the first available FTP server.

DISCLAIMER: The appearance of any external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Affairs (VA) of this Web site or the information, products, or services contained therein. The VA does not exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of this VA Intranet Service.

I. User Guide

This is the User Guide section of this supplemental documentation for the BSE-related software. It is intended for use in conjunction with the BSE Project. It details the user-related BSE documentation (e.g., overview of the BSE Project, management of the BSE-related software, etc.).

1. Broker Security Enhancement (BSE) Project Overview

This chapter describes the mechanism by which the Broker Security Enhancement (BSE) enables RPC Broker Delphi-based applications to make remote user/visitor connections in a more secure manner. This new BSE-based mechanism will subsequently replace the current Compensation And Pension Records Interchange (CAPRI)-based mechanism for remote user/visitor access by RPC Broker Delphi-based client/server applications.

Introduction

The Veterans Health Administration (VHA) information systems management and user community has expressed a need to secure access to patient information at remote sites.

Some VistA application users require access to data located at remote sites at which the users:

- Do *not* have assigned Access and Verify codes.
- Have *not* been entered into the NEW PERSON file (#200) by Information Resource Management (IRM) staff.
- Want to avoid having multiple Access/Verify code pairs.

The Compensation And Pension Records Interchange (CAPRI) application was the first application with these requirements. This application is used by Veterans Benefits Administration (VBA) staff to remotely access VistA data related to claims for veterans treated at any VistA site.

The CAPRI application was the first application to use the modified version of the VistA Remote Procedure Call (RPC) Broker software, which was based on the Remote Data Views (RDV) access method, as a means for obtaining such access. This access enters the user's information into the NEW PERSON file (#200) as a visitor, but does *not* require an Access or Verify code for the user at the remote site. As a result of the CAPRI application, there has been an increase in the number of other applications that also require or are requesting this type of remote data access.

The goal of the Broker Security Enhancement (BSE) Project is to accomplish the following:

- Enable RPC Broker Delphi-based applications to access Remote VistA M Servers with increased security.
- Enhance the RPC Broker method used to connect to Remote VistA M Servers.
- Ensure correct information for user access to prevent the mistaken identification of an incorrect or non-existent user (spoofing) via unauthorized applications.
- Provide the ability for RPC Broker Delphi-based applications that have implemented BSE to specify their own context option.
- Allow the VistA Imaging Display Client to pull in images from remote sites without requiring credentials on the Remote VistA M Servers.

Features

The Broker Security Enhancement (BSE) Project provides the following features and functionality:

- Adds a step to the RPC Broker signon process to authenticate the connecting application. This will also involve passing a secret encoded phrase that will be established on the VistA M Server via a patch and KIDS build.
- Adds a step to the RPC Broker signon process on the Remote VistA M Server to authenticate the user by connecting back to the Authenticating VistA M Server.
- Provides the capability for remote applications to specify their own context option.

Architectural Scope

The architectural scope of the BSE Project is as follows:

- **Use of Kernel Authentication—**Kernel is used as the authenticator. Kernel is a valid means of authenticating on a backend VistA M Server.
- **Client/Server-based Application Support—**This document only discusses the BSE functionality provided with VistA RPC Broker Delphi-based client/server applications.

Process Overview

The overall process to make a remote connection via an RPC Broker Delphi-based client/server application that has implemented the Broker Security enhancement (BSE) is as follows:

1. The user starts the BSE-enabled application.
2. The BSE-enabled application connects to the Authenticating VistA M Server and presents the VistA login GUI dialogue to the user.

   NOTE: The Authenticating VistA M Server is identified in the CALLBACKSERVER field (#.03) in the CALLBACKTYPE Multiple field (#1) in the REMOTE APPLICATION file (#8994.5).

3. The user enters their Kernel Access and Verify codes, is authenticated via Kernel, and is signed onto the BSE-enabled application's Authenticating VistA M Server.
4. The BSE-enabled application gets a Kernel Authentication Token for the authenticated user from the Authenticating VistA M Server. This token will eventually be used by the Remote VistA M Server to obtain the necessary user information for populating a user as a "visitor" entry in the remote site's NEW PERSON file (#200). This ensures the following:

   - The user will *not* be spoofed.
   - The data at the remote site is valid.

   A sample Kernel Authentication Token follows:

   ```
   XWBHDL977-124367_0
   ```

5. The BSE-enabled application completes any other processing necessary to identify the Remote VistA M Server and gathers any other required information.
6. The BSE-enabled application disconnects from the Authenticating VistA M Server.
7. The BSE-enabled application performs the following tasks:
   a. Creates a Security Pass Phrase value that is composed of the following two pieces of data:
      - **Security Phrase**—A one-way hashed value that is stored in the REMOTE APPLICATION file (#8994.5) and used to identify the BSE-enabled application's file entry.

        REF: For more information on the Security Phrase, please refer to the "Security Phrase" topic in Chapter 2, "BSE-related VistA Applications and Modules," in this manual.

      - **Kernel Authentication Token**

   b. Sets the SecurityPhrase property of the RPCBroker login component to the Security Pass Phrase value (see Step #7a), which will later be used by the Remote VistA M Server to call back the Authenticating VistA M Server.

   c. Sets the other appropriate RPCBroker login component properties in order to call the Remote VistA M Server.

REF: For more information on the specific RPCBroker login component property settings, please refer to the "Step-By-Step Procedures to Implement BSE" topic in Chapter 4, "Implementing BSE in VistA RPC Broker-based Applications," in this manual.

8. The BSE-enabled application connects to the Remote VistA M Server with the RPCBroker login component passing in the encoded value of the SecurityPhrase property (see Step #7).

CAUTION: Remote access is only permitted at sites that have installed the application's information (including the hashed Security Phrase) into the REMOTE APPLICATION file (#8994.5), ensuring that a rogue application *cannot* obtain access.

REF: For more information on the application's Security Phrase, please refer to the "Security Phrase" topic in Chapter 2, "BSE-related VistA Applications and Modules," in this manual.

9. The Kernel software on the Remote VistA M Server performs the following tasks:
   a. Identifies and hashes the decoded value of the RPCBroker login component's SecurityPhrase property (see Step #7a and b).
   b. Uses the hashed value of the BSE-enabled application's Security Pass Phrase to identify the application's entry in the REMOTE APPLICATION file (#8994.5).

NOTE: Included in that entry is the mechanism(s) for contacting the Authenticating VistA M Server.

c. Connects to the Authenticating VistA M Server passing in the Kernel Authentication Token that identifies the user.

d. Obtains the user demographic information from the Authenticating VistA M Server. This user demographic information is used to establish the user as a remote user/visitor.

e. Disconnects from the Authenticating VistA M Server.

f. Uses the demographic information obtained from the Authenticating VistA M Server to set up the user as a visitor entry on the Remote VistA M Server. It creates or matches an entry in the NEW PERSON file (#200) and provides the visitor with the context option specified for the BSE-enabled application in the REMOTE APPLICATION file (#8994.5).

10. The BSE-enabled application is notified by the RPCBroker login component that it successfully connected and that the user is signed on to the Remote VistA M Server. The user can then continue with any processing necessary on the Remote VistA M Server. If for some reason the user signon fails on the Remote VistA M Server, the user will be prompted to enter a valid Access and Verify code on the Remote VistA M Server. If the user cancels the signon, he/she is prompted with a signon cancellation dialogue box.

REF: For more information on the REMOTE APPLICATION file (#8994.5), please refer to the "REMOTE APPLICATION File (#8994.5)" topic in Chapter 2, "BSE-related VistA Applications and Modules," in this manual.

If any of the following error conditions exist, the user is prompted with a regular GUI signon dialogue instructing them to enter their Access and Verify codes:

- No entry for the application in the REMOTE APPLICATION file (#8994.5).
- No match for the Kernel Authentication Token.
- Cannot connect to the Authenticating VistA M Server.

The Remote VistA M Server connects to the Authenticating VistA M Server and passes in the Kernel Authentication Token identifying the user. The Authenticating VistA M Server responds back by returning the demographic information necessary to establish the user as a remote user. The Remote VistA M Server disconnects from the Authenticating VistA M Server and sets up the user's profile as a visitor entry, including the necessary context option specified for the application in the REMOTE APPLICATION file (#8994.5).

The BSE-enabled application is notified that the user is signed on and continues processing as normal.

There are basically two classes of applications that will use this BSE authentication mechanism:

Application Class	Description
Single Server Authentication	Applications that require users to authenticate against a single VistA M Server and determine the remote locations to be accessed (e.g., CAPRI). For those applications where the users all authenticate on a single VistA M Server, the application only needs to specify the Uniform Resource Locator (URL) for its VistA M Server and one or more methods for connecting to it (including port number[s]) in the CALLBACKTYPE Multiple of the REMOTE APPLICATION file (#8994.5).
Multiple Server Authentication	Applications that require users to authenticate at their local medical center or other site (e.g., VistAWeb or other Web-based applications). For those applications where each user authenticates on multiple/different VistA M Servers, the application needs to obtain both a Kernel Authentication Token and the demographic data necessary for identifying or adding a remote user/visitor during the authentication process on the Authenticating VistA M Server. The application passes in the Kernel Authentication Token and application Security Pass Phrase, as described above (see the "Process Overview" topic). The REMOTE APPLICATION file (#8994.5) contains an address for the Web-based application and the Remote VistA M Server returns the Kernel Authentication Token to the application and expects it to return the demographic information associated with that Kernel Authentication Token. This requires the application to keep the Kernel Authentication Token and demographic data in a location that is accessible by the application until the demographic data has been provided to the Remote VistA M Server.  **RECOMMENDATION: The Infrastructure & Security Services (ISS) BSE Development Team highly encourages that other non-Web-based applications use a single server rather than multiple servers for user authentication.**

Table 1-1. BSE Project—Application authentication server class types

Process Diagrams

The following diagram illustrates the BSE process sequence flow:

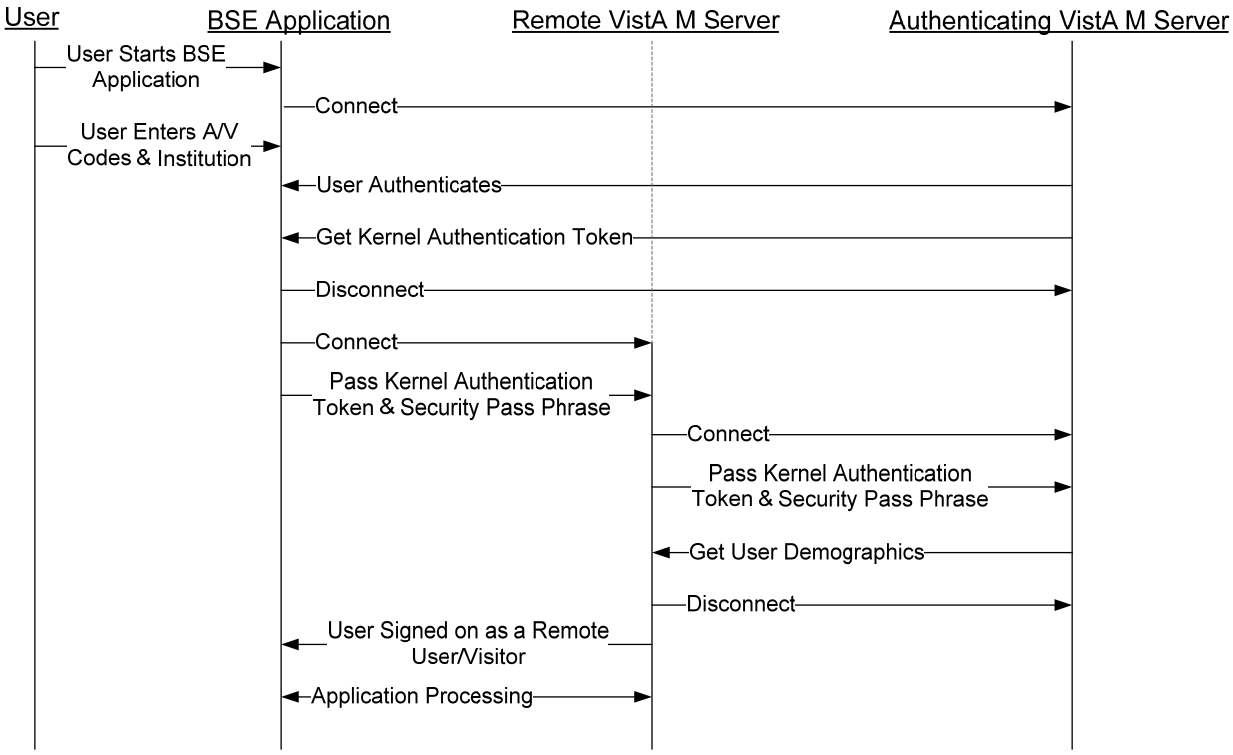

Figure 1-1. BSE Project—Process sequence flow

The following diagram illustrates the BSE process overview:

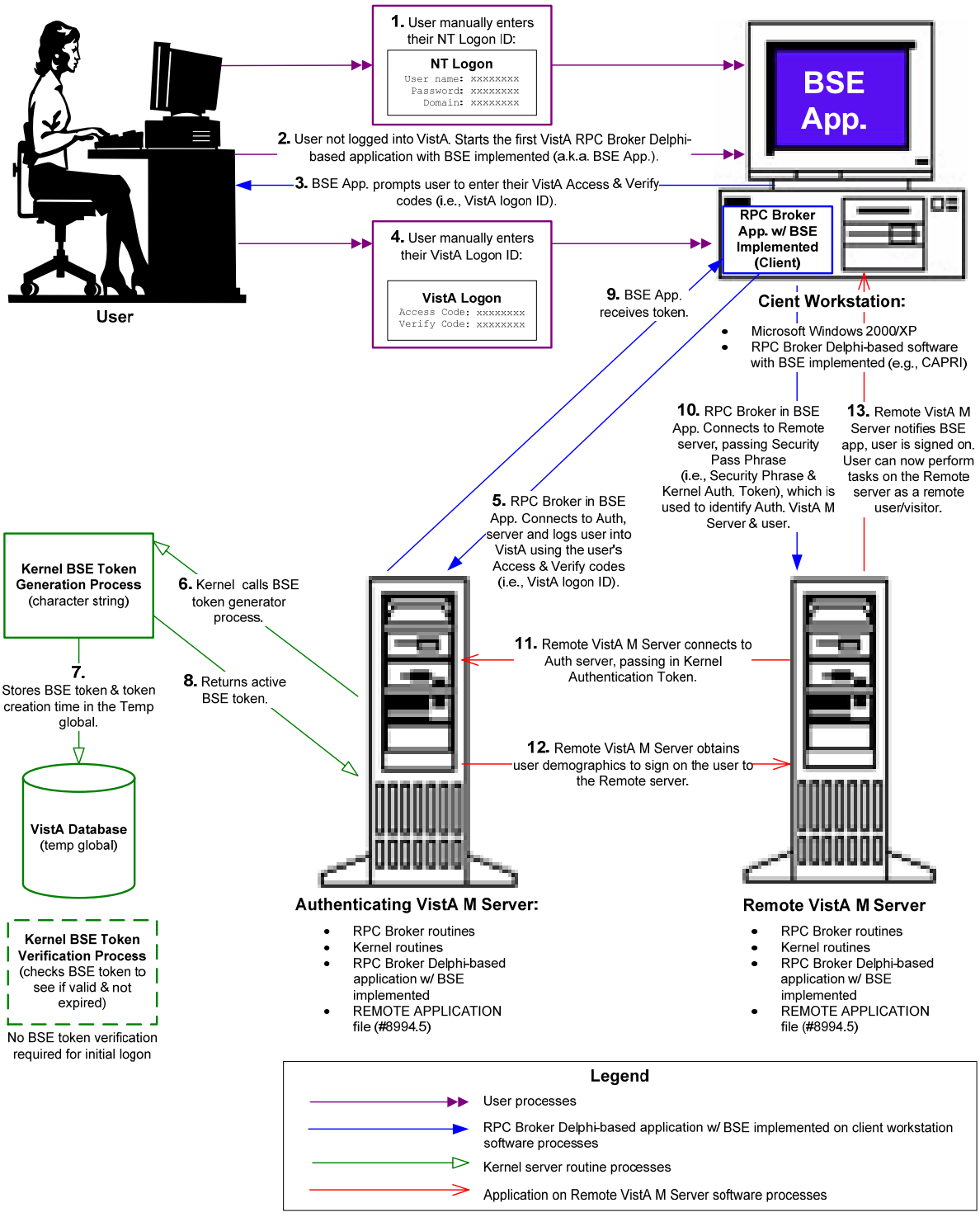

Figure 1-2. BSE Project—Process overview

2. BSE-related VistA Applications and Modules

The chapter describes the new or modified functionality made to the BSE-related software applications and modules as listed in Table 2-1.

An RPC Broker Delphi-based and BSE-enabled VistA application comprises software that has been re-compiled using the RPC Broker login component, modified for BSE. BSE capability comes into play when you are using a BSE-enabled application (e.g., Compensation And Pension Records Interchange [CAPRI] or VistAWeb).

REF: For information on how to implement BSE in VistA RPC Broker Delphi-based client/server applications, please refer to Chapter 4, "Implementing BSE in VistA RPC Broker-based Applications," in this manual.

This manual discusses in more detail the various software applications and modules that, together, provide for BSE functionality:

Application/Module	Location	Description
VistA M Server	VistA M Server	This is the "backend server" where the Kernel and RPC Broker software act as the authentication source for all VistA applications (i.e., client/server, rich client, Web, and roll-and-scroll applications). The VistA M Server also executes remote procedure calls (RPCs) and provides other functions to VistA applications. **REF:** For a list of BSE-related Vista M Server patches, please refer to Table 3-2.
Client/Server Login Component: RPC Broker	Client (Developer workstations only)	The RPCBroker login component allows client/server applications to authenticate against the VistA M Server and obtain a persistent connection over which remote procedure calls (RPCs) are executed. This component is modified in the BSE Project to be more secure when accessing data at remote sites. RPC Broker-based applications using remote or visitor access (e.g., Compensation And Pension Records Interchange [CAPRI], VistAWeb) *must* invoke this modified RPC Broker login component to implement the Broker Security Enhancement (BSE).

Table 2-1. BSE Project—Software applications and modules

REF: For the specific software patches required for the implementation of BSE, please refer to Table 3-2 in this chapter.

Kernel—Authentication Interface to VistA

Authentication is the process of verifying a user identity to ensure that the person requesting access to a VistA system (e.g., clinical information system) is, in fact, that person to whom entry is authorized.

Currently, Kernel on the VistA M Server is the approved method to provide both Authentication and Authorization (AA) services for all VistA applications Kernel was assessed as the most straightforward and timely approach to also be used for remote signon authentication in the BSE Project. By using Kernel as the authenticator for BSE, the NEW PERSON file (#200) continues to serve as the single user data store for VistA and the BSE Project.

Some potential advantages to employing Kernel as the AA source include the following:

- Ease of file maintenance by IRM.
- Provides a single point of user management for existing and new VistA RPC Broker Delphi-based applications.
- Allows the use of an existing credential (i.e., the Access and Verify code) for Authentication and Authorization, rather than introducing a new security credential.
- Ease of coding requirements by application developers.
- Avoids an additional user store, which simplifies the migration to any future AA solutions.

The BSE functionality for Kernel was introduced with Kernel Patch XU*8.0*404 (server-side). The BSE functionality includes the creation of a Kernel Authentication Token. The Kernel Authentication Token is generated once a user has been initially authenticated on the Authenticating VistA M Server via their Access and Verify codes. This Kernel Authentication Token can then be used to authenticate a user on a Remote VistA M Server.

RPC Broker

The RPC Broker software consists of both a client and server software piece.

Client

The RPC Broker login component is embedded in a Borland Delphi-based rich client/server application (e.g., Compensation And Pension Records Interchange [CAPRI]). The RPCBroker login component is used to connect the application running on a Microsoft Windows client workstation to the VistA M Server. This connection allows data retrieval from the VistA M Server database. The RPCBroker login component uses Kernel's Access and Verify codes to authenticate a user to VistA.

The BSE functionality for the RPCBroker login component was introduced with RPC Broker Patch XWB*1.1*45 (client-side) and Kernel Patch XU*8.0*404 (server-side). BSE functionality includes the addition of a new property to the RPCBroker login component that allows applications to pass an application's Security Phrase and Kernel Authentication Token, which is referred to in this documentation as the Security Pass Phrase. Thus, when a VistA RPC Broker Delphi-based application, such as CAPRI, is recompiled with the BSE-updated RPCBroker login component and other required code modifications are made, that application would then become capable of accessing Remote VistA M Servers without requiring users to re-enter their Access and Verify codes.

Server

In order to implement BSE and use the RPC-Broker callback type, the central Authenticating VistA M server *must* run the RPC Broker as a TCPIP service. The Non-callback RPC Broker Listener/TCPIP service is distributed and described with RPC Broker Patch XWB*1.1*35 and updated with XWB*1.1*44.

REF: For more information on the RPC Broker and TCPIP service setup, please refer to RPC Broker Patches XWB*1.1*35 and 44 on FORUM and the RPC Broker documentation, specifically the *RPC Broker TCP/IP Supplement*, located on the VDL at the following Web address:

http://www.va.gov/vdl/Infrastructure.asp?appID=23

REF: For more detailed information on the application developer procedures and code modifications needed to implement BSE in RPC Broker Delphi-based applications, please refer to Chapter 4, "Implementing BSE in VistA RPC Broker-based ," in this manual.

REMOTE APPLICATION File (#8994.5)

The REMOTE APPLICATION file (#8994.5) was released with RPC Broker Patch XWB*1.1*45. This file helps better secure remote user/visitor access to Remote VistA M Servers initiated by RPC Broker Delphi-based GUI applications. Remote user/visitor access permits applications where users need to access a large number of sites and do so *without* requiring separate Access and Verify codes at each target remote site.

The REMOTE APPLICATION file (#8994.5) contains the following fields:

Field Name	Field Number	Description
NAME	.01	(required) This is the name for the RPC Broker Delphi-based application that requires remote user/visitor access. The name must be from 3 to 30 characters, not numeric or starting with punctuation.
CONTEXTOPTION	.02	(required) This is the name of the context option (i.e., client/server or "B"-type option) that the application users will need. The name must be from 3 to 45 characters. The user is signed on as a visitor and given this context option as a secondary menu option.
APPLICATIONCODE	.03	(required) This is the hashed value for an application's Security Phrase. **REF:** For more information on the Security Phrase, please refer to the "Security Phrase" topic that follows in this chapter.

Field Name	Field Number	Description
CALLBACKTYPE	1	(required) This is a Multiple field. It can contain multiple values describing the mechanisms by which the Remote VistA M Server can contact the application's Authenticating VistA M Server to obtain the demographic information. It consists of the subfields described below.
CALLBACKTYPE (CALLBACKTYPE Multiple)	.01	(required) This field indicates the mechanism(s) by which the server should contact the Authenticating VistA M Server to obtain information necessary to sign the current user onto the current server. The values for this field are: • R—RPC Broker TCP/IP connection • M—M-to-M Broker connection • H—HyperText Transport Protocol (HTTP) connection
CALLBACKPORT (CALLBACKTYPE Multiple)	.02	(required) This is the port number (3 – 5 characters) to be used for the callback connection to the Authenticating VistA M Server for the CALLBACKTYPE (#.01) specified.
CALLBACKSERVER (CALLBACKTYPE Multiple)	.03	(required) This is the server designation (address) to be used for the callback to the Authenticating VistA M Server for the CALLBACKTYPE (#.01) specified. This should be a Domain Name Service (DNS) name-based address rather than an Internet Protocol (IP) address, because IP addresses can change. It should be a server name ending in MED.VA.GOV or MED.VHA.VA.GOV. The DNS servers resolve the name, and thus, ensure that the site is a valid VistA M Server.

URLSTRING (CALLBACKTYPE Multiple)	.04	(optional) This field holds the text that should follow the SERVER ADDRESS (field #.03) for HTTP connections to obtain the information for the Kernel Authentication Token passed in for a REMOTE APPLICATION connection.
		If the complete Uniform Resource Locator (URL) to be used for the callback is:
		`http://myserver.med.va.gov/some/kind/of/location/somePage.aspx`
		The CALLBACKSERVER (#.03) field could be:
		`myserver.med.va.gov`
		and the URLSTRING would be:
		`some/kind/of/location/somePage.aspx`
		This field is only used if the CALLBACKTYPE filed (#.01) value is H for HTTP.

Table 2-2. Fields in the REMOTE APPLICATION file (#8994.5)

REF: For more information on the REMOTE APPLICATION file (#8994.5), please refer to the "Files" topic in Chapter 6, "Implementation and Maintenance," in this manual.

Security Phrase

The Security Phrase is an RPC Broker Delphi-based application's entry into the REMOTE APPLICATION file (#8994.5). The Security Phrase is a general phrase that is known only to the application that created it. When it is stored in the REMOTE APPLICATION file (#8994.5), it *must* be hashed. This one-way hashed value, which is the result of a call to the $$EN^XUSHSH(phrase) API, is entered into the APPLICATIONCODE field (#.03) in the REMOTE APPLICATION file (#8994.5) for the application.

This Security Phrase is combined with the Kernel Authentication Token to make up the Security Pass Phrase, which is then stored in the SecurityPhrase property of the RPCBroker login component.

CAUTION: It is important to realize that the Security Phrase identifies only those applications that are authorized to perform remote user/visitor access. Thus, the stored value of the Security Phrase is a one-way hash so that other rogue applications *cannot* mimic an application and access the Remote VistA M Server.

RECOMMENDATION: Since the Security Phrase is the application's identifier, the Infrastructure & Security Services (ISS) BSE Development Team recommends developers identify the Security Phrase as a const value in an include file in any RPC Broker Delphi-based program implementing BSE. A substitute include file containing a phrase similar to the Security Phrase should then be included with release of the source code.

Kernel Authentication Token

The Kernel Authentication Token is generated by the same code used to generate handles (i.e., a unique text string that is used to identify a specific user for which it was generated) for other purposes used in the RPC Broker software. Once created, the token is stored in the ^XTMP temporary global. The basic format of the token (handle) is as follows:

XWBHDLnnn-nnnnnn_n

The "XWBHDL" indicates that it is an RPC Broker handle, where "XWB" is the RPC Broker namespace and "HDL" indicates that it is a handle.

The following is an example of a Kernel Authentication Token:

```
XWBHDL977-124367_0
```

II. Developer Guide

This is the Developer Guide section of this supplemental documentation for the BSE-related software. It is intended for use in conjunction with the BSE Project. It details the programmer-related BSE documentation (e.g., developer procedures needed to make RPC Broker Delphi-based applications BSE-enabled, APIs exported with the BSE-related software, etc.).

3. BSE Installation Instructions for Developers

Preliminary Considerations: Developer Workstation Requirements

The following minimum hardware and software tools/utilities are required by developers when developing RPC Broker Delphi-based client/server applications that implement BSE:

Minimum Hardware/Software Requirement	Description
Workstation Hardware	80x86-based client or server workstation.
Operating System	One of the following 32-bit operating systems: • Microsoft Windows XP • Microsoft Windows 2000
Development-related Software	The following development-related software is required, in order to develop RPC Broker Delphi-based client/server applications that implement BSE: • BSE-related software—BSE client software, see Table 2-1 and Table 3-2. • Borland Delphi Integrated Development Environment (IDE)—Commercial-Off-The-Shelf (COTS) software for development of RPC Broker/COM-based rich client applications. • RPC Broker Development Kit (BDK) V. 1.1 (i.e., RPC Broker Patch XWB*1.1*40) for development of COM-based client/server applications. • Updated RPC Broker files (i.e., those files released with RPC Broker Patch XWB*1.1*45, see Table 3-4).  **REF:** For more information on configuring files and integrating BSE functionality with RPC Broker Delphi-based client/server software applications, please refer to Chapter 4, "Implementing BSE in VistA RPC Broker-based Applications," in this manual.

Minimum Hardware/Software Requirement	Description
Network Communications Software/Capability **REF:** For more information on telecommunications support, please visit the VA Office of Information and Technology (OIT) Home Page: http://vaww.va.gov/oirm/telecom/	All developer client or server workstations *must* have the following network communications software and capability: • Networked client/server workstations running Microsoft's native TCP/IP stack. **NOTE:** Currently, only Winsock compliant TCP/IP protocol is supported on the LAN or remotely as Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP). You *must* use RAS (Remote Access Service) or Dialup Networking to connect to the server using PPP or SLIP. For the setup of RAS or Dialup Networking, please refer to the appropriate operating system's documentation. • Connectivity with the VistA M Server (i.e., VA Wide Area Network [WAN] connectivity). Run PING.EXE to test the connectivity. • Capability to log onto the NT network using a unique NT Logon ID.

Table 3-1. BSE Project—Developer minimum hardware and software tools/utilities required for RPC Broker Delphi-based and BSE-enabled application development

Dependencies—VistA M Server Patches

The Broker Security Enhancement (BSE)-related software comprises two patches and software releases from the following VistA applications (listed by category and software name):

Category	Software	Version	Patch	Subject/Description
Server	Kernel	8.0	XU*8.0*404	Broker Security Enhancement (BSE)—This VistA M Server patch helps better secure remote user/visitor access to Remote VistA M Servers initiated by RPC Broker Delphi-based GUI applications. Remote user/visitor access permits applications where users need to access a large number of sites and do so *without* requiring a separate Access and Verify code at each site. In order to use this functionality, developers must enable their RPC Broker-based client/server applications with the Broker Security Enhancement (BSE). This patch works in conjunction with RPC Broker Patch XWB*1.1*45.
	RPC Broker	1.1	XWB*1.1*45	Broker Security Enhancement (BSE)—This client and server-side patch contains the modified RPCBroker login component units to help better secure remote user/visitor access to Remote VistA M Servers initiated by RPC Broker Delphi-based GUI applications. Remote user/visitor access permits applications where users need to access a large number of sites and do so *without* requiring a separate Access and Verify code at each site. This patch works in conjunction with Kernel Patch XU*8.0*404. RPC Broker Patch XWB*1.1*45 *must* be installed before Kernel Patch XU*8.0*404.
Client (for develop-ment)	RPC Broker	1.1	XWB*1.1*40	You will need to first install the latest RPC Broker Development Kit (BDK). This software is not distributed with the Broker Security Enhancement (BSE) Project but is available with RPC Broker Patch XWB*1.1*40 (i.e., xwb1_1p40pg.exe). This software/patch is available on the Enterprise VistA Support (EVS) anonymous directories or on the RPC Broker Download Web site at the following Web address: http://vaww.vista.med.va.gov/broker/download.asp

Category	Software	Version	Patch	Subject/Description
			XWB*1.1*45	Broker Security Enhancement (BSE)—This client and server-side patch contains the modified RPCBroker login component units to help better secure remote user/visitor access to Remote VistA M Servers initiated by RPC Broker Delphi-based GUI applications. Remote user/visitor access permits applications where users need to access a large number of sites and do so *without* requiring a separate Access and Verify code at each site. This patch works in conjunction with Kernel Patch XU*8.0*404. RPC Broker Patch XWB*1.1*45 *must* be installed before Kernel Patch XU*8.0*404. The XWB1_1P45.zip file contains only those BDK files that were modified for the BSE Project, which includes changes to some of the files originally distributed with the xwb1_1p40pg.exe self-installing executable file. **NOTE:** This client-side software is dependent on the server-side RPC Broker Patch XWB*1.1*45 and Kernel Patch XU*8.0*404.

Table 3-2. BSE Project—VistA M Server patch dependencies

REF: For specific VistA M Server patch details, please refer to the Patch Module on FORUM.

NOTE: This table only includes VistA M Server software patches required for BSE; it does *not* list COTS software or other VistA software/patches that are not directly related to BSE.

BSE VistA M Server Installation Instructions

The installation instructions in this section are directed at the Information Resource Management (IRM) personnel located at a site and are applicable for the Test/Production accounts in a Caché environment.

1. Confirm/Obtain VistA M Server Distribution Files *(recommended)*

The following files are needed to install the BSE Project-related VistA M Server software:

File Name	Type	Description
Xwb1_1p45-readme.txt	ASCII	**Readme File** (manual). This file provides any pre-installation instructions, last minute changes, new instructions, and additional information to supplement the manuals. Read all sections of this file prior to following the installation instructions in this manual. .
XWB1_1P45_ SUPPLEMENTAL.PDF	Binary	**Supplement to Patch Description** (manual). Use in conjunction with the Readme text file (i.e., Xwb1_1p45-readme.txt).
XU_8_P404.KID	ASCII	**Kernel Patch XU*8.0*404.** KIDS build for Kernel Patch XU*8.0*404 (see Table 3-2 for patch details). Follow normal procedures to obtain and install this released patch (see FORUM).
XWB1_1P45.KID	ASCII	**RPC Broker Patch XWB*1.1*45.** KIDS build for RPC Broker Patch XWB*1.1*35 (see Table 3-2 for patch details). Follow normal procedures to obtain and install this released patch (see FORUM).

Table 3-3. Distribution files—BSE Project-related VistA M Server files

REF: For the BSE Project software release, all distribution files, unless otherwise noted, are available for download from the Enterprise VistA Support (EVS) anonymous directories:

- Albany OIFO ftp://ftp.fo-albany.med.va.gov/
- Hines OIFO ftp://ftp.fo-hines.med.va.gov/
- Salt Lake City OIFO ftp://ftp.fo-slc.med.va.gov/
- Preferred Method download.vista.med.va.gov

  This method transmits the files from the first available FTP server.

2. Retrieve VistA M Server Patches *(required)*

At the time of publication of this manual, two VistA M Server-side patches are required for BSE Project installation (see Table 3-2). You should have these patches readily available so that you can apply them later in the installation process. Obtain all released BSE Project-related VistA M Server-side patches from the Patch module on FORUM or through normal procedures.

NOTE: The Broker Security Enhancement (BSE)-related software comprises two patches and software releases from the following VistA applications (listed alphabetically):

- Kernel—Kernel Patch XU*8.0*404
- RPC Broker—RPC Broker Patch XWB*1.1*45

REF: For the specific VistA M Server software patches required for the implementation of BSE, please refer to Table 3-2 in this chapter.

3. Do Not Run any RPC Broker Client/Server Software with BSE Implemented During the Installation *(required)*

No RPC Broker Delphi-based software with BSE implemented or M-to-M Broker-based applications should be running while the BSE Project-related software installation on the VistA M Server is taking place.

4. Verify KIDS Install Platform *(required)*

Verify that the Kernel Installation and Distribution System (KIDS) platform on your system is ready to install VistA M Server patches.

A. Verify Host File Server (HFS) Device in the DEVICE File (#3.5)

Verify that you have a Host File Server (HFS) device in the DEVICE file (#3.5) named **"HFS"**. If you have performed KIDS installations on the VistA M Server before, you probably already have an appropriate HFS device set up. If you don't have an entry for this device, you *must* create one.

REF: For information on how to create an HFS device, please refer to Chapter 18, "Host Files," in the *Kernel Systems Manual*.

B. Verify Null Device in the DEVICE File (#3.5)

Verify that you have a Null device in the DEVICE file (#3.5) named "NULL" (or whose mnemonic is named "NULL").

You can have other devices with similar names, but one device is needed whose name or mnemonic is "NULL." The subtype should be a "P-" subtype (e.g., P-OTHER), the margin should be a minimum of 80, and the page length should be a minimum of 60. Sample setups:

Caché for OpenVMS Null Device Setup Example

```
NAME: NULL                          $I: _NLA0:
  ASK DEVICE: NO                    ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO         LOCATION OF TERMINAL: Bit Bucket
  SUBTYPE: P-OTHER                  TYPE: TERMINAL
```

Caché/NT Null Device Setup Example

```
NAME: NULL                          $I: //./nul
  ASK DEVICE: NO                    ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO         LOCATION OF TERMINAL: BIT BUCKIT
  SUBTYPE: P-OTHER                  TYPE: TERMINAL
```

P-OTHER Terminal Type Setup Example

```
NAME: P-OTHER                       RIGHT MARGIN: 132
  FORM FEED: #                      PAGE LENGTH: 64
  BACK SPACE: $C(8)                 DESCRIPTION: General prntr (132)
```

5. Install BSE Project-related VistA M Server Patches *(required)*

The BSE Project-related VistA M Server patches are listed in Table 3-2. All VistA M Server patches are distributed in Kernel V. 8.0 KIDS format. Follow the normal procedures to obtain released patches.

Make sure that the Kernel and RPC Broker software is fully patched. Patches *must* be installed in their published sequence.

Using KIDS, load and install the BSE Project-related VistA M Server patches on all VistA M systems to which any RPC Broker Delphi-based application with BSE implemented will be connecting (i.e., VistA M Server Test and Production accounts).

Installers *must* follow the patch installation instructions under the "Installation Instructions" section in the patch descriptions for Kernel Patch XU*8.0*404 and RPC Broker Patch XWB*1.1*45.

REF: For more information on these patches, please refer to Table 3-2 in this chapter or the Patch Module on FORUM.

REF: For more information on KIDS, please refer to the KIDS section in the *Kernel Systems Manual* located on the VDL at the following Web address:

http://www.va.gov/vdl/Infrastructure.asp?appID=10

6. Run the RPC Broker as a TCPIP Service *(required)*

In order to implement BSE and use the RPC-Broker callback type, the central Authenticating VistA M server *must* run the RPC Broker as a TCPIP service. The Non-callback RPC Broker Listener/TCPIP service is distributed and described with RPC Broker Patch XWB*1.1*35 and updated with XWB*1.1*44.

REF: For more information on the RPC Broker and TCPIP service setup, please refer to RPC Broker Patches XWB*1.1*35 and 44 on FORUM and the RPC Broker documentation, specifically the *RPC Broker TCP/IP Supplement*, located on the VDL at the following Web address:

http://www.va.gov/vdl/Infrastructure.asp?appID=23

Congratulations! You have now completed the installation of BSE Project-related software on the VistA M Server.

BSE Client Workstation Installation Instructions

The following instructions are only required for those client workstations to be used by programmers to develop RPC Broker Delphi-based client/server software applications and wish to implement the Broker Security Enhancement (BSE).

DISCLAIMER: These installation instructions assume that the client workstation already has an earlier version of the RPC Broker BDK software installed (e.g., xwb1_1p40pg.exe).

REF: For Developer Client Workstation platform requirements, please refer to the "Preliminary Considerations: Developer Workstation Requirements" topic in this chapter.

1. Confirm/Obtain Developer Client Workstation Distribution Files *(recommended)*

The following files are needed to install the BSE-related Developer Client Workstation software:

File Name	Type	Description
Xwb1_1p45-readme.txt	ASCII	**Readme File** (manual). This file provides any pre-installation instructions, last minute changes, new instructions, and additional information to supplement the manuals. Read all sections of this file prior to following the installation instructions in this manual.
XWB1_1P45_ SUPPLEMENTAL.PDF	Binary	**Supplement to Patch Description** (manual). Use in conjunction with the Readme text file (i.e., Xwb1_1p45-readme.txt).
XWB1_1P45.zip (see RPC Broker Patch XWB*1.1*45 on FORUM)	Binary	**RPC Broker/Delphi-based Programmer Client Workstation** (client software). This zip file contains the following files: • BSE sample files: – BseSample1.cfg – BseSample1.dof – BseSample1.dpr – BseSample1.exe – BseSample1.inc – BseSample1.res – BSEWebServer.exe – fBseSample1.dfm – fBseSample1.jpg

File Name	Type	Description
		– fBseSample1.pas • BSE source files: – Loginfrm.pas – RpcSLogin.pas – Trpcb.pas • ReadMe file

Table 3-4. BSE Project—Distribution files for developer client workstations

2. Create a BSE Staging Folder *(required)*

Create a BSE Staging Folder on your Developer Client Workstation. This will be referred to as the **<STAGING_FOLDER>** for the rest of the instructions.

3. Download and Unzip the XWB1_1P45.zip File *(required)*

A. Download the XWB1_1P45.zip File

Download the XWB1_1P45.zip file into the **<STAGING_FOLDER>** created in Step #2.

B. Unzip the XWB1_1P45.zip File

After unzipping the XWB1_1P45.zip file, the following directory structure will be created containing the various RPCBroker login component units:

```
<STAGING_FOLDER>
|
|--BDK32
   |
   |--Samples
   |  |
   |  |--BSE
   |  |  |
   |  |  |--BseSample1.cfg
   |  |  |--BseSample1.dof
   |  |  |--BseSample1.dpr
   |  |  |--BseSample1.exe
   |  |  |--BseSample1.inc
   |  |  |--BseSample1.res
   |  |  |--BSEWebServer.exe
   |  |  |--fBseSample1.dfm
   |  |  |--fBseSample1.jpg
   |  |  |--fBseSample1.pas
   |  |
   |  |--XWB1_1P45-README.TXT
   |
   |--Source
      |
      |--Loginfrm.pas
      |--RpcSLogin.pas
      |--Trpcb.pas
```

Figure 3-1. Directory structure of the RPCBroker login component units related to implementing BSE

4. Backup the RPC Broker BDK32 Directory *(recommended)*

Navigate to the RPC Broker BDK directory. BDK directories may vary from workstation to workstation, because users can select different directory locations during the initial BDK installation process. The default installation directory is as follows:

```
C:\Progam Files\Vista\BDK32
```

We recommend that you backup (copy) all of the original files in this BDK directory to a safe location prior to replacing or adding any new BSE-related files during the installation process.

5. Update the RPC Broker BDK32 Files *(required)*

A. Replace Updated BDK32 Source Files

Navigate to the BDK32 Source directory (..\BDK32\Source). Replace all of the following source files with the source files from the BSE distribution file (i.e., XWB1_1P45.zip, see Step #2):

- Loginfrm.pas
- RpcSLogin.pas
- Trpcb.pas

B. Add the BSE Sample Directory

Navigate to the BDK32 Samples directory (..\BDK32\Samples). Add the BSE sample directory and subordinate files from the BSE distribution file (i.e., XWB1_1P45.zip, see Step #2).

REF: For more information on integrating BSE with VistA client/server-based software applications, please refer to Chapter 4, "Implementing BSE in VistA RPC Broker-based Applications," in this manual.

Congratulations! You have now completed the installation of the BSE-related software on the *Developer Client Workstation*.

4. Implementing BSE in VistA RPC Broker-based Applications

This chapter describes how application developers can modify their VistA RPC Broker Delphi-based client/server applications in order to implement the Broker Security Enhancement (BSE). The following topics are discussed:

- Assumptions When Implementing BSE
- Step-By-Step Procedures to Implement BSE

Assumptions When Implementing BSE

The following assumptions are made regarding application developers and VistA software applications when implementing BSE:

- **Developer Training—**Application developers should already be knowledgeable/trained in creating RPC Broker Delphi-based applications.
- **RPC Broker-based Applications—**RPC Broker Delphi-based application already exists.
- **Login at Startup—**Applications automatically initiate login at application startup (i.e., users are presented with an Access/Verify login dialogue).
- **VistA M Server Patches—**All BSE Project-related VistA M Server patches have been loaded on the appropriate servers.

Step-By-Step Procedures to Implement BSE

This topic describes the procedures that VistA application developers *must* follow in order to implement the Broker Security Enhancement (BSE) in their RPC Broker Delphi-based applications (i.e., COM client applications developed in Borland Delphi), so that the application can make remote user/visitor connections.

NOTE: This new BSE-based mechanism will subsequently replace the current CAPRI-based mechanism for remote user/visitor access by RPC Broker Delphi-based client/server applications.

1. Create a Unique Application Security Phrase *(required)*

Use the $$EN^XUSHSH API to create a hashed Security Phrase (case sensitive) that is unique for your application.

For example, in M, go to Programmer Mode and enter the following command:

```
W $$EN^XUSHSH("My Application Security Phrase")
```

The resulting sample value is:

```
\9U_s]WMY[tS{ffL~s7Q
```

CAUTION: This is a sample value only; do *not* use this as your Application Security Phrase!

This is a one-way hash value for the Security Phrase that is only known to the application that creates it.

RECOMMENDATION: Since the Security Phrase is the application's identifier, the Infrastructure & Security Services (ISS) BSE Development Team recommends developers identify the Security Phrase as a const value in an include file in any RPC Broker Delphi-based program implementing BSE. A substitute include file containing a phrase similar to that used above should then be included with release of the source code.

REF: For more information on the application Security Phrase, please refer to the "Security Phrase" topic in Chapter 2, "BSE-related VistA Applications and Modules," in this manual.

2. Create an Entry in the REMOTE APPLICATION File (#8994.5) *(required)*

An application *must* send out a patch that creates an entry for their RPC Broker Delphi-based application that has implemented BSE in the REMOTE APPLICATION file (#8994.5). Developers *must* add entries to the following fields in File #8994.5:

- NAME (#.01)—Enter a descriptive name for your application.
- CONTEXTOPTION (#.02)—Enter the name of the "B"-Type context option that the users will need to run the application.
- APPLICATIONCODE (#.03)—Enter the hashed value of the Security Phrase you created in Step #1.
- CALLBACKTYPE Multiple (#1):
  - CALLBACKTYPE (#.01)—Current values for this field are:
    - R—RPC Broker TCP/IP connection.
    - M—M-to-M Broker connection.
    - H—HyperText Transport Protocol (HTTP) connection communication. You *must* also add an entry in the URLSTRING field (#.04).
  - CALLBACKPORT (#.02)—Enter the Port number to be used for the callback connection.
  - CALLBACKSERVER (#.03)—Enter the address of the server to be used for the callback connection. This should be a Domain Name Service (DNS) name-based address rather than an Internet Protocol (IP) address, because IP addresses can change.
  - URLSTRING (#.04)—Used only if the CALLBACKTYPE field (#.01) contains H for HyperText Transport Protocol (HTTP). Enter the Uniform Resource Locator (URL) string for the callback to the HTTP server.

NOTE: For more information on the REMOTE APPLICATION file (#8994.5) and specific field entries, please refer to the "REMOTE APPLICATION File (#8994.5)" topic in Chapter 2, "BSE-related VistA Applications and Modules," in this manual.

3. Include Updated RPCBroker Login Component Units *(required)*

RPC Broker Delphi-based applications use an RPCBroker login component to connect to the VistA M Server.

To implement BSE into your VistA RPC Broker Delphi-based client/server applications, developers *must* include the updated RPCBroker login component units, which are located in the ..\BDK32\Source directory (see Figure 3-1), prior to compiling the updated version of the application.

REF: Use the sample code in the BseSample1.pas file when implementing BSE in your application. The BseSample1.pas file is located in the ..\BDK32\Samples\BSE directory (see Figure 3-1).

```
procedure TForm1.DoConnection(Key: String);
var
  TokenValue: String;
begin
  RPCB.Server := AuthServer.Text;
  RPCB.ListenerPort := StrToInt(AuthPort.Text);
  RPCB.Connected := True;
  if RPCB.Connected then
  begin
    RPCB.RemoteProcedure := 'XUS SET VISITOR';
    RPCB.Call;
    TokenValue := RPCB.Results[0];
    RPCB.Connected := False;
    ShowMessage('Token: '+TokenValue);
    if not (TokenValue = '') then
    begin
      RPCB.Server := RemoteServer.Text;
      RPCB.ListenerPort := StrToInt(RemotePort.Text);
      RPCB.SecurityPhrase := Key + '^' + TokenValue;
      RPCB.Connected := True;
      if RPCB.Connected then
      begin
        ShowMessage('Signed  on to Remote Server');
        RPCB.CreateContext('XWB BROKER EXAMPLE');
        btnDisconnect.Enabled := True;
        btnEcho.Enabled := True;
        btnM2M.Enabled := False;
        btnTCPIP.Enabled := False;
      end
      else
        ShowMessage('Connection to Remote Server failed!');
    end;
  end
  else
    ShowMessage('Initial Sign-on Failed');
end;
```

Figure 4-1. fBseSample1.pas file—Sample code excerpt (#1)

4. Get Kernel Authentication Token *(required)*

After authenticating the user, the Kernel software on the Authenticating VistA M Server creates the Kernel Authentication Token. Kernel stores this token in the ^XTMP temporary global.

The application calls the XUS SET VISITOR RPC to get the Kernel Authentication Token for the user. This token is then passed to the RPC Broker component and used to create a Security Pass Phrase (see Step #5 that follows). This token will eventually be used to obtain the necessary user information for populating a user as a "visitor" entry in the remote site's NEW PERSON file (#200).

```
RPCB.Server := AuthServer.Text;
RPCB.ListenerPort := StrToInt(AuthPort.Text);
RPCB.Connected := True;
if RPCB.Connected then
begin
  RPCB.RemoteProcedure := 'XUS SET VISITOR';
  RPCB.Call;
  TokenValue := RPCB.Results[0];
  RPCB.Connected := False;
```

Figure 4-2. fBseSample1.pas file—Sample code excerpt (#2)

5. Create and Encode a Security Pass Phrase *(required)*

The application creates a Security Pass Phrase (string). The Security Pass Phrase consists of the application Security Phrase (Step #1) concatenated with he Kernel Authentication Token delimited by a caret ("^"). For example:

```
My Application Security Phrase^XWBHDL977-124367_0
```

The RPC Broker software encodes the Security Pass Phrase, which will be passed to the Remote VistA M Server for authentication.

In the source code excerpt that follows (see Figure 4-3), the value of **Key** (i.e., constant) was defined earlier by importing an include file that contained the following two lines:

```
const
    Key = ' My Application Security Phrase';
```

NOTE: Key is a constant, which is a type of variable that has a fixed value that *cannot* be changed.

This is the code after the RPCBroker login component connection to the Authenticating VistA M Server has been disconnected:

```
    if not (TokenValue = '') then
    begin
      RPCB.Server := RemoteServer.Text;
      RPCB.ListenerPort := StrToInt(RemotePort.Text);
      RPCB.SecurityPhrase := Key + '^' + TokenValue;
      RPCB.Connected := True;
      if RPCB.Connected then
      begin
        ShowMessage('Signed  on to Remote Server');
        RPCB.CreateContext('XWB BROKER EXAMPLE');
        btnDisconnect.Enabled := True;
        btnEcho.Enabled := True;
        btnM2M.Enabled := False;
        btnTCPIP.Enabled := False;
      end
      else
        ShowMessage('Connection to Remote Server failed!');
    end;
  end
```

Figure 4-3. fBseSample1.pas file—Sample code excerpt (#3)

6. Set RPCBroker Login Component Properties *(required)*

The developer *must* set the following RPCBroker login component properties when calling the Remote VistA M Server:

- Server—This property is set to the Domain Name Service (DNS) or Internet Protocol (IP) address of the Remote VistA M Server.
- ListenerPort—This property is set to the Listener Port number of the Remote VistA M Server.
- (new) SecurityPhrase property—This property is set to the unhashed application's Security Phrase concatenated with the Kernel Authentication Token (See Step #5).
- Connect—This property is set to True.

7. Process Remote User/Visitor Login on Remote Server *(required)*

After connecting to the Remote VistA M Server, software running on the Remote VistA M Server will do the following:

A. Identify the Security Pass Phrase

Kernel identifies the data passed in as a parameter, which contains the application's Security Phrase and Kernel Authentication Token for the user.

B. Hash the Security Pass Phrase

Kernel hashes the Security Pass Phrase to parse out the application's Security Phrase and the Kernel Authentication Token.

C. Get Authenticating VistA M Server Connection Mechanism

Kernel uses the Security Phrase to identify the application's entry in the REMOTE APPLICATION file (#8994.5).

Included in that entry is the mechanism(s) for contacting the Authenticating VistA M Server (i.e., the connection type [R—RPC Broker TCP/IP connection, M—M-to-M Broker, or H—HyperText Transport Protocol connection], port number, and address [IP, DNS, or URL]).

NOTE: The mechanism(s) for contacting the Authenticating VistA M Server allows you to use either the IP address or DNS; however, the ISS BSE Development Team recommends that you use the DNS.

D. Connect to Authenticating Server Using Kernel Authentication Token

The Remote VistA M Server uses the appropriate mechanism to identify and connect to the Authenticating VistA M Server, passing in the Kernel Authentication Token that identifies the user.

E. Obtain User Demographics

Kernel uses the XUS GET VISITOR RPC to request and obtain the user demographic information from the Authenticating VistA M Serve.

The user demographic information that is returned is the same authenticated user information that is returned with the Remote Data Views (RDV) software
(i.e., ssn^name^station name^station number^DUZ^phone^vpid):

- Social Security Number (SSN)
- Name
- Station Name
- Station Number
- DUZ

- Telephone
- VA-wide Person Identifier (VPID)

This user demographic information will be used to later establish the user as a remote user/visitor on the Remote VistA M Server.

F. Disconnect from the Authenticating VistA M Server

The Remote VistA M Server disconnects from the Authenticating VistA M Server.

G. Set Up User as a Visitor Entry on the Remote VistA M Server

Kernel uses the demographic information obtained from the Authenticating VistA M Server to set up the user as a visitor entry on the Remote VistA M Server.

Kernel creates or matches an entry in the NEW PERSON file (#200) and provides the visitor with the context option specified for the application in the REMOTE APPLICATION file (#8994.5).The matching process is the same process used with the Remote Data Views (RDV) software.

8. Recompile Your Application *(required)*

Developers *must* recompile their RPC Broker Delphi-based application when all edits/updates are completed.

9. Test Your Application *(recommended)*

Developers should test their RPC Broker Delphi-based applications to ensure they have successfully implemented BSE.

BSE Sample Test Applications

The Broker Security Enhancement (BSE) client software distribution file (i.e., XWB1_1P45.zip) includes the BrokerSecurityEnhancement Sample1 application (i.e., BseSample1.exe, see Figure 4-4) and Broker Security Enhancement Test WebServer (i.e., BSEWebServer.exe, see Figure 4-7).

You can use these sample applications to help test the sample entries in the REMOTE APPLICATION file (#8994.5) and to test the different connection types (i.e., TCP/IP, M2M, and/or HTTP) to verify that the VistA M Server-side is set up correctly to implement BSE.

CAUTION: In order to implement BSE and use the RPC-Broker callback type, the central Authenticating VistA M server *must* run the RPC Broker as a TCPIP service.

BrokerSecurityEnhancement Sample1

The sample application dialogue provided by the BrokerSecurityEnhancement Sample1 application (i.e., BseSample1.exe) is shown below:

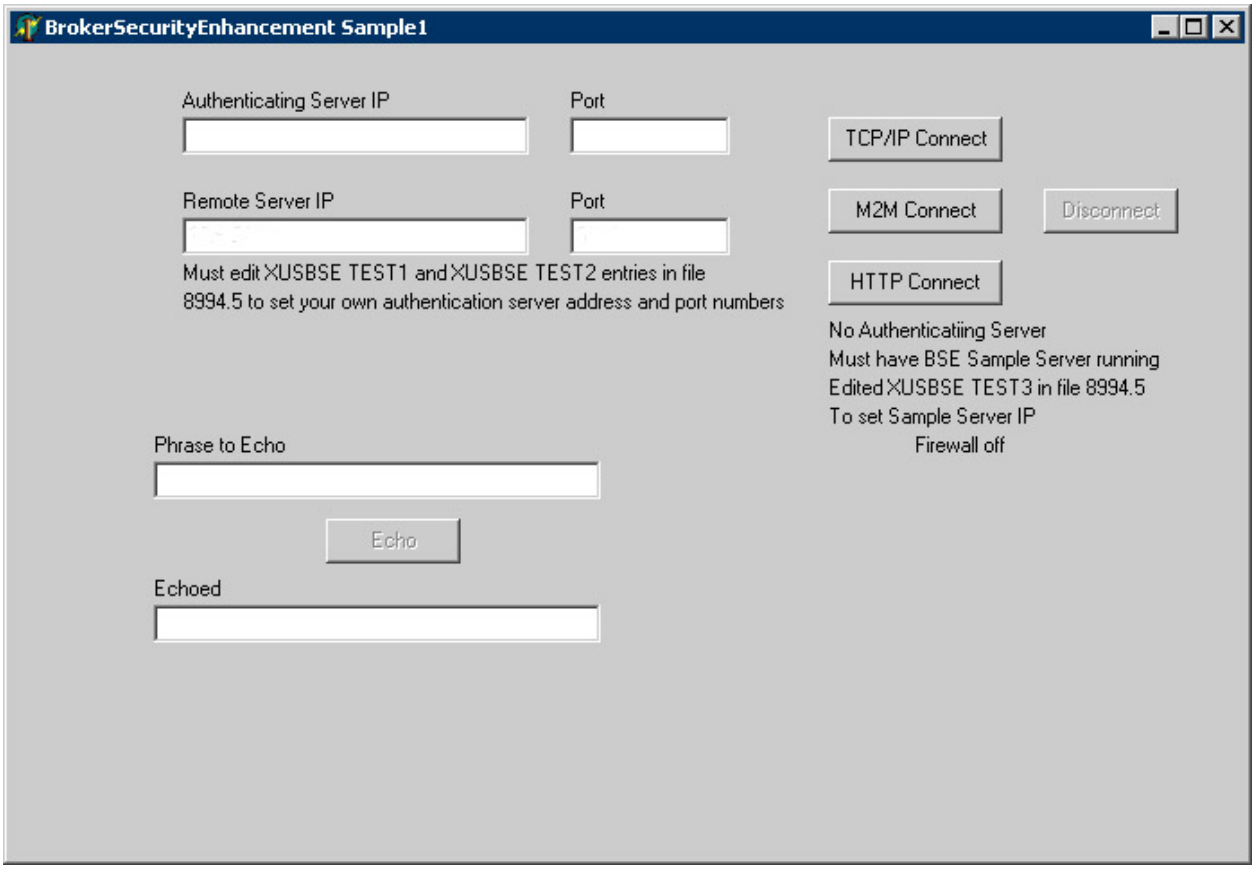

Figure 4-4. BSE Project—BrokerSecurityEnhancement Sample1 application (i.e., BseSample1.exe)

The sample application has the following controls:

- Server Edit Fields:
  - **Authenticating Server IP—**IP address for the Authenticating VistA M Server. This field is empty at initial startup; it is an editable field.
  - (Authenticating Server) **Port—**Port number for the Authenticating VistA M Server. This field is empty at initial startup; it is an editable field.
  - **Remote Server IP—**IP address for the Remote VistA M Server. This field is empty at initial startup; it is an editable field.
  - (Remote Server) **Port—**Port number for the Remote VistA M Server. This field is empty at initial startup; it is an editable field**.**

NOTE: Both the Authenticating and Remote VistA M Servers *must* have RPC Broker Patch XWB*1.1*45 and Kernel Patch XU*8*404 installed.

- Connection Buttons:
  - **TCP/IP Connect**
  - **M2M Connect**
  - **HTTP Connect**
  - **Disconnect**
- Phrase Echo Controls:
  - **Phrase to Echo** Edit Field—Enter an echo phrase.
  - **Echo** Button—Button used to submit the phrase to be echoed back form the Remote VistA M Server.
  - **Echoed** Field—Contains the phrase that gets echoed back once the user/visitor is signed onto the Remote VistA M Server.

To successfully run and test the BrokerSecurityEnhancement Sample1 application (i.e., BseSample1.exe), do the following:

A. Edit entries for XUSBSE TEST1 and XUSBSE TEST2 in the following fields in the CALLBACKTYPE Multiple (#1) in the REMOTE APPLICATION file (#8994.5). These entries will be the Authenticating VistA M Servers that are used to authenticate the current user, and to which a callback will be made to obtain information to eventually create the visitor entry in the Remote VistA M Server:

   - CALLBACKPORT (#.02)
   - CALLBACKSERVER (#.03)

   The Broker Security Enhancement (BSE)-related code is dependent upon the use of appropriate and valid information for the Authenticating and Remote VistA M Servers. Therefore, running the BseSample1.exe program requires that you populate these fields on the Remote VistA M Server.

   The Authenticating VistA M Server is the server on which the user already has a valid Kernel Access and Verify code established (i.e., entry in the NEW PERSON file [#200]). Both the Authenticating and Remote VistA M Servers *must* also have RPC Broker Patch XWB*1.1*45 and Kernel Patch XU*8*404 installed.

B. Start the BseSample1.exe program.

C. Enter a valid Authenticating VistA M Server IP address and Port number.

   NOTE: This is the server against which the user first authenticates.

D. Enter a valid Remote VistA M Server IP address and Port number.

NOTE: This will be the server that the user will sign onto as a visitor (and already contains the updated information for the Authenticating VistA M Server in the REMOTE APPLICATION file ([#8994.5]).

E. Click on (press) one of the connection buttons (e.g., **TCP/IP Connect** button).

REF: For more information on using the **HTTP Connect** button, please refer to the "HTTP and WebServer Functionality" topic that follows in this chapter.

F. Enter Access and Verify codes in the VistA Sign-on dialogue box when prompted.

NOTE: This authenticates the user against the Authenticating VistA M Server.

G. (optional) Choose your Division (i.e., Station Number) to log into, if prompted.

H. Click on (press) the **OK** button when presented with the following dialogue:

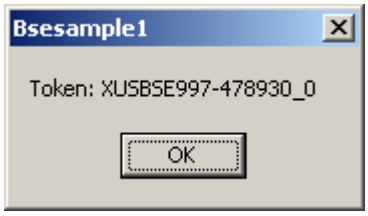

Figure 4-5. Sample Kernel Authentication Token

NOTE: This dialogue indicates that the Kernel Authentication Token was created, which means the user is now authenticated on the Authenticating VistA M Server.

I. After a few moments, you will get another dialogue confirming the user is now also authenticated on the Remote VistA M Server as a visitor. Click on (press) the **OK** button when presented with the following dialogue:

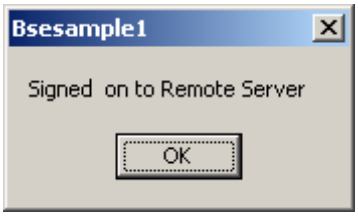

Figure 4-6. Sample confirmation message indicating the user is signed onto the Remote VistA M Server as a visitor

J. You can now enter an echo phrase to the Remote VistA M Server and get the string echoed back.

HTTP and WebServer Functionality

The HTTP functionality is currently tested by running the BSEWebServer.exe program.

If you are going to use the **HTTP Connect** button, the XUSBSE TEST3 entry *must* also have the address of a workstation running the BSEWebServer.

The sample application dialogue provided by the Broker Security Enhancement Test WebServer application (i.e., BSEWebServer.exe) is shown below:

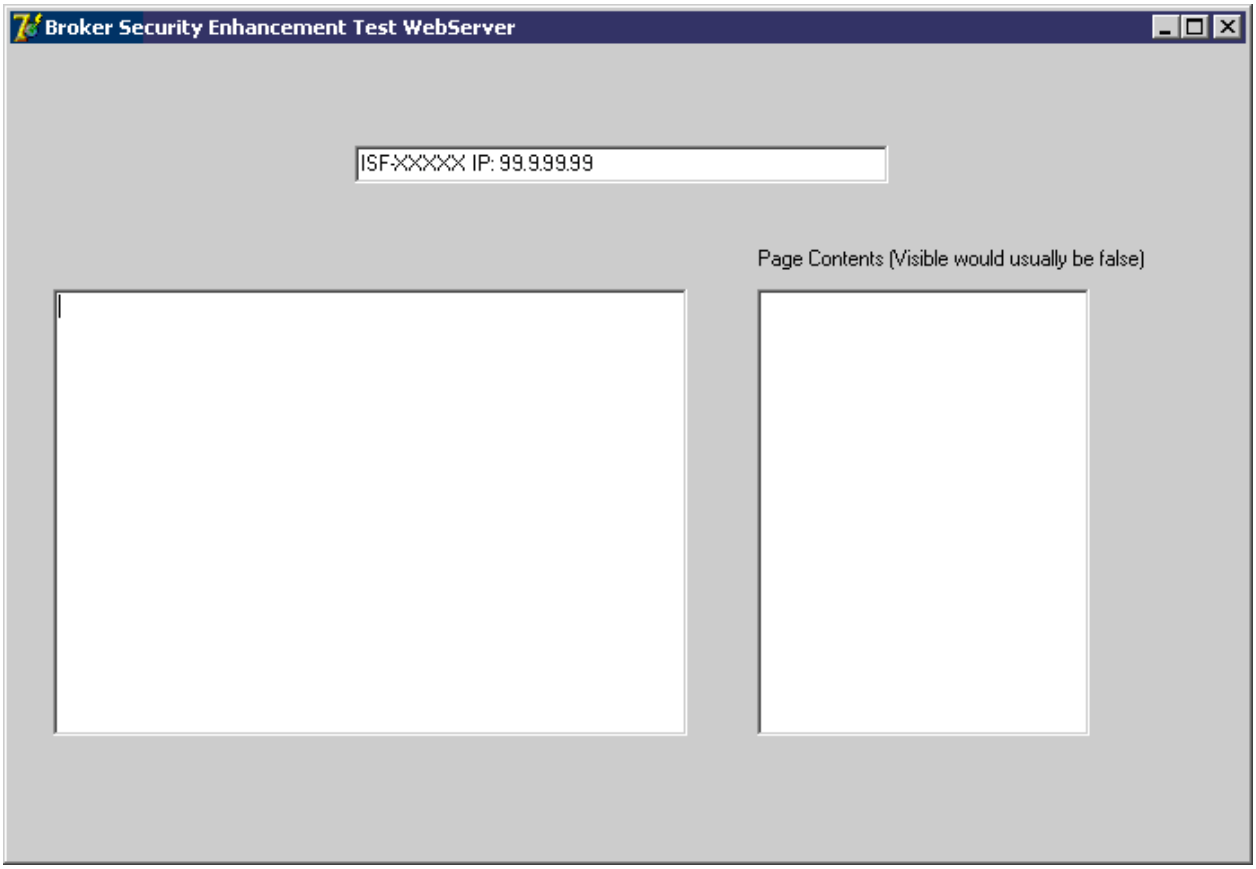

Figure 4-7. BSE Project—Broker Security Enhancement Test WebServer application (i.e., BSEWebServer.exe)

When the application starts, it will show the client workstation name and its current IP address. Edit the entry for XUSBSE TEST3 in the REMOTE APPLICATION file (#8994.5). This IP address should be entered in response to the prompt for the CALLBACKSERVER field (#.03) in the CALLBACKTYPE Multiple field (#1). The CALLBACKPORT field (#.02) can be set to 80, and the URLSTRING field (#.04) can be left blank or set to a forward slash character ("/"). This BSEWebServer application currently only looks for a data element with the character string "xVAL=" and returns a fixed set of dummy data, which can be signed on as a visitor. In a real application, the token would be returned to the Web server (e.g., VistaWeb), and it would return the user data obtained from the Authenticating VistA M Server for that user.

After being notified that you have been connected to the Remote VistA M Server, type in a character string and press the **Echo** button. The text string should come back to you.

5. Application Program Interfaces (APIs)

There are no new or modified *public* Kernel or RPC Broker APIs distributed with the BSE Project-related software (i.e., RPC Broker Patch XWB*1.1*45.

NOTE: Other *private* Kernel and RPC Broker APIs are used internally by the RPC Broker server-side login module and are *not* documented here.

III. Systems Management Guide

This is the Systems Management Guide section of this supplemental documentation for the BSE-related software. It is intended for use in conjunction with the BSE Project. It details the technical-related BSE documentation (e.g., implementation and maintenance of the BSE-related software, routines, files, options, interfaces, product security, etc.).

6. Implementation and Maintenance

Information throughout this manual is meant to help Information Resource Management (IRM) personnel in the implementation and maintenance of the Broker Security Enhancement (BSE)-related software.

Namespace

The BSE-related software consists of patches that have been assigned to the following namespaces (listed alphabetically):

- XU—Kernel
- XWB—RPC Broker

NOTE: The Broker Security Enhancement (BSE)-related software comprises two patches and software releases from the following VistA applications (listed alphabetically):

- Kernel—Kernel Patch XU*8.0*404
- RPC Broker—RPC Broker Patch XWB*1.1*45

REF: For the specific VistA M Server software patches required for the implementation of BSE, please refer to Table 3-2 in Chapter 3, "BSE Installation Instructions for Developers" in this manual.

Remote Procedure Calls (RPCs)

The following *public* remote procedure call (RPC) is exported with the BSE-related software:

RPC Name	RPC Description
XUS SET VISITOR RPC	(public) This RPC was added with the BSE Project (i.e. added with Kernel Patch XU*8.0*404). This RPC is run on the Authenticating VistA M Server. It returns a Kernel Authentication Token that is used to identify the current user. Tag^Routine: SETVISIT^XUSBSE1

Table 6-1. BSE Project—RPCs

NOTE: Other *private* Kernel and RPC Broker RPCs are used internally by the RPC Broker server-side login module and are *not* documented here.

Files

The following file is exported with the BSE-related software:

File Number	File Name	Field Description
8994.5	REMOTE APPLICATION	This file was introduced as part of the Broker Security Enhancement (BSE) Project. It was released with RPC Broker Patch XWB*1.1*45. This file helps better secure remote user/visitor access to Remote VistA M Servers initiated by RPC Broker Delphi-based GUI applications. Remote user/visitor access permits applications where users need to access a large number of sites and do so *without* requiring a separate Access and Verify code at each site. Once BSE is fully implemented, those RPC Broker Delphi-based applications that require remote/visitor access *must* have an entry in this file with a one-way hash of a secure phrase. It is a one-way hash value that is only known to the application that creates it. Identification of an entry in the file is based on the application passing in the original phrase which is then hashed and used for a cross-reference lookup. The application must have at least one entry in the CALLBACKTYPE Multiple field (#1) indicating all of the following: • Connection type • Valid address for the authenticating server • Connection port number • URL String for HTTP connections This information is necessary for the Remote VistA M Server to directly connect the Authenticating VistA M Server to obtain the demographic information necessary to create or match the user/visitor entry in the NEW PERSON file (#200). The application will also specify the desired context option for the user/visitor. This will be given to the remote user/visitor instead of forcing the application to determine how to set this value. **NOTE:** This will be a very small file, containing only entries for those applications that will use the Broker Security Enhancement (BSE) for remote visitor's access mechanism. The total number of entries should be well under 20.

Figure 6-1. BSE Project—Files

Global Mapping/Translation, Journaling, and Protection

There are *no* special global mapping/translation, journaling, and protection instructions for the BSE-related software.

Routines

This topic contains a list of the new or modified routines exported with the BSE-related software. A brief description of the routines is provided.

Software	Routine Name	Routine Description
Kernel	XUSBSE1	This routine was released with Kernel Patch XU*1.1*404. It contains various functions and procedures that are used by the Broker Security Enhancement (BSE) Project.
	XUSBSE2	This routine was released with Kernel Patch XU*1.1*404. It contains various functions and procedures that are used by the Broker Security Enhancement (BSE) Project.
	XUSBSEUT	This routine was released with Kernel Patch XU*1.1*404. It is the Broker Security Enhancement (BSE) Project unit test routine.
	XUSRB	This routine was modified and released with Kernel Patch XU*1.1*404. It contains various functions and procedures that are used by the Broker Security Enhancement (BSE) Project.
RPC Broker	XWB45PO	This routine was released with RPC Broker Patch XWB*1.1*45. It contains various functions and procedures that are used by the Broker Security Enhancement (BSE) Project.
	XWBM2MEZ	This routine was released with RPC Broker Patch XWB*1.1*45. It contains various functions and procedures for M-to-M Broker server connections that are used by the Broker Security Enhancement (BSE) Project.
	XWBRM	M2M Broker Server Request Manager. This routine was enhanced with RPC Broker Patch XWB*1.1*45 for the Broker Security Enhancement (BSE) Project.
	XWBTCPM2	Test WEB Service. This routine was enhanced with RPC Broker Patch XWB*1.1*45 for the Broker Security Enhancement (BSE) Project.

Table 6-2. BSE Project—Routines

Exported Options

There are *no* options exported with the BSE-related software.

Archiving and Purging

There are *no* special archiving or journaling instructions for the BSE-related software.

Callable Routines

There are *no public* callable routine entry points (i.e., Application Program Interfaces [APIs]) distributed with the BSE-related software.

External Relations

VistA Software Requirements

The BSE software relies on the following VistA software to run effectively (listed alphabetically):

Software	Version	Software/Patch Information
Kernel	8.0	Server software—Fully patched.
RPC Broker	1.1	Client/Server software—Fully patched.

Table 6-3. BSE Project—External Relations with VistA software

NOTE: The Broker Security Enhancement (BSE)-related software comprises two patches and software releases from the following VistA applications (listed alphabetically):

- Kernel—Kernel Patch XU*8.0*404
- RPC Broker—RPC Broker Patch XWB*1.1*45

REF: For the specific VistA M Server software patches required for the implementation of BSE, please refer to Table 3-2 in Chapter 3, "BSE Installation Instructions for Developers" in this manual.

NOTE: There are *no* other COTS (*non*-VA) products embedded in or requiring special interfaces by this version of the BSE-related software, other than those provided by the underlying operating systems.

Integration Agreements (IAs)

The Database Administrator (DBA) maintains a list of Integration Agreements (IAs) or mutual agreements between software developers allowing the use of internal entry points or other software-specific features that are not available to the general programming public.

The BSE-related software is *not* dependent on any agreements.

To obtain the current list of IAs to which Kernel or RPC Broker is a custodian:

1. Sign on to the FORUM system (forum.va.gov).
2. Go to the DBA menu [DBA].
3. Select the Integration Agreements Menu option [DBA IA ISC].

4. Select the Custodial Package Menu option [DBA IA CUSTODIAL MENU].
5. Choose the ACTIVE by Custodial Package option [DBA IA CUSTODIAL].
6. When this option prompts you for a package, enter **XXXX—**Where **XXXX** equals: **XU** or **Kernel**; or **XWB** or **RPC Broker**.
7. All current IAs to which the software is a custodian are listed.

To obtain detailed information on a specific integration agreement:

1. Sign on to the FORUM system (forum.va.gov).
2. Go to the DBA menu [DBA].
3. Select the Integration Agreements Menu option [DBA IA ISC].
4. Select the Inquire option [DBA IA INQUIRY].
5. When prompted for "INTEGRATION REFERENCES," enter the specific integration agreement number of the IA you would like to display.
6. The option then lists the full text of the IA you requested.

To obtain the current list of IAs to which Kernel or RPC Broker is a subscriber:

1. Sign on to the FORUM system (forum.va.gov).
2. Go to the DBA menu [DBA].
3. Select the Integration Agreements Menu option [DBA IA ISC].
4. Select the Subscriber Package Menu option [DBA IA SUBSCRIBER MENU].
5. Choose the Print ACTIVE by Subscribing Package option [DBA IA SUBSCRIBER].
6. When prompted with "START WITH SUBSCRIBING PACKAGE," enter **XXXX** (in uppercase). When prompted with "GO TO SUBSCRIBING PACKAGE," enter **XXXX** (in uppercase)—Where "**XXXX**" equals: **XU** or **XWB**.
7. All current IAs to which the software is a subscriber are listed.

Internal Relations

Relationship of the BSE-related software with VistA

Namespace

The BSE-related software consists of patches that have been assigned to the following namespaces (listed alphabetically):

- XU—Kernel
- XWB—RPC Broker

NOTE: The Broker Security Enhancement (BSE)-related software comprises two patches and software releases from the following VistA applications (listed alphabetically):

- Kernel—Kernel Patch XU*8.0*404
- RPC Broker—RPC Broker Patch XWB*1.1*45

REF: For the specific VistA M Server software patches required for the implementation of BSE, please refer to Table 3-2 in Chapter 1, "Broker Security Enhancement (BSE) Project Overview" in this manual.

Kernel V. 8.0

In order to develop RPC Broker-based applications so that they are BSE-enabled, the Kernel Server Patch XU*8.0*404 *must* be installed.

RPC Broker V. 1.1

In order to develop RPC Broker-based applications so that they are BSE-enabled, the RPC Broker Server Patch XWB*1.1*45 *must* be installed.

Software-wide and Key Variables

The BSE-related software does *not* employ the use of software-wide or key variables.

SACC Exemptions

The BSE-related software does *not* have any Programming Standards and Conventions (SAC) exemptions.

7. Software Product Security

Security Management

There are *no* special legal requirements involved in the use of the BSE-related software.

Each RPC Broker Delphi-based application that implements BSE must create a unique Security Phrase that is a one-way hash that gets stored in the APPLICATIONCODE field (#.03) in the REMOTE APPLICATION file (#8994.5).

REF: For more information on the Security Phrase, please refer to the "Security Phrase" topic in Chapter 2, "BSE-related VistA Applications and Modules," in this manual.

CAUTION: Any information or component related to this software that is considered to be of sensitive information (e.g., hash routines) will *not* be included in any Freedom of Information Act (FOIA) request releases.

Mail Groups, Bulletins, and Alerts

Mail Groups

The BSE-related software does *not* create or utilize any specific mail groups.

Bulletins

The BSE-related software does *not* create any bulletins.

Alerts

The BSE-related software does *not* create any alerts.

Remote Systems

The RPC Broker login component connects and transmits data to Local and Remote VistA M Servers using Transmission Control Protocol/Internet Protocol (TCP/IP), allowing connections between client workstation RPC Broker Delphi-based applications and the VistA M Server. VistA RPC Broker Delphi-based applications can use any remote procedure call (RPC) authorized to the application, if the application is authorized to the signed-on user.

Interfaces

The Broker Security Enhancement (BSE) Project does *not* have its own user interface for GUI applications. This project will, however, create developer or site configuration screens.

There are *no* COTS (*non*-VA) products embedded in or requiring special interfaces by this version of the BSE Project-related software, other than those provided by the underlying operating systems.

Applications that use the BSE for remote visitors will need to make a couple of changes in their applications. Applications that use the security enhanced RPC Broker utility are responsible for creating their own user interfaces.

Electronic Signatures

There are *no* electronic signatures used within the BSE-related software.

Security Keys

There are *no* specific security keys exported with this version of the BSE-related software.

File Security

There following file security is associated with the BSE-related software.

Number	Name	DD	RD	WR	DEL	LAYGO	AUDIT
8994.5	REMOTE APPLICATION	@	@	@	@	@	@

Table 7-1. BSE Project—File Security

Official Policies

There are *no* special legal requirements involved in the use of the BSE-related software.

Distribution of the BSE-related software is unrestricted.

As per the Software Engineering Process Group/Software Quality Assurance (SEPG/SQA) Standard Operating Procedure (SOP) 192-039—Interface Control Registration and Approval (effective 01/29/01), application programmers *must* not alter any Health*e*Vet VistA Class I software code.

REF: For more information on SOP 192-039—Interface Control Registration and Approval, please refer to the following Web address:

http://vista.med.va.gov/SEPG_lib/Standard%20Operating%20Procedures/192-039%20Interface%20Control%20Registration%20and%20Approval.htm

Glossary

CAPRI	The **C**ompensation **A**nd **P**ension **R**ecords **I**nterchange software was designed to promote efficient communications between the Veterans Health Administration (VHA) and the Veterans Benefits Administration (VBA). It offers VBA Rating Veteran Service Representatives and Decision Review Officers help in building the rating decision documentation through online access to medical data. It creates a more efficient means of requesting compensation and pension examinations and navigating existing patient records.
CART-CL	The **C**ardiac **A**ssessment **R**eporting and **T**racking **S**ystem – **C**ath **L**ab (CART-CL) is third party software under development by medical center staff in Seattle and Denver. It is a collaborative effort among Patient Care Services, Office of Information (OI), Office of Quality and Performance, and the Ischemic Heart Disease Quality Enhancement Research Initiative (IHD-QUERI). The goal of CART-CL is to support standardized data capture and generation of pre-procedure and procedure reports for all VA cardiac catheterization labs. CART-CL will provide a mechanism to capture procedure information in a single national database that conforms to American College of Cardiology's National Cardiac Catheterization Data Repository (ACC-NCDR). Furthermore, CART-CL will send procedure information to Event Capture/PCE for inclusion in VA national databases.
CBO	**C**hief **B**usiness **O**ffice (CBO)—Effort for Centralizing Fiscal Process (Jerry Sicard Requestor)
CLIENT	A single term used interchangeably to refer to the user, the workstation, and the portion of the program that runs on the workstation. This term is typically used in an object-oriented environment, where a client is a member of a group that uses the services of an unrelated group. If the client is on a local area network (LAN), it can share resources with another computer (server). With respect to the BSE software, client refers to the "requesting server" that is able to connect to a "receiving server," where both servers reside in VistA on the same or on different VistA M systems.
COMPONENT	An object-oriented term used to describe the building blocks of GUI RPC Broker Delphi-based Delphi applications. A software object that contains data and code. A component may or may not be visible. These components interact with other components on a form to create the GUI user application interface.
CPRS	**C**omputerized **P**atient **R**ecord **S**ystem
CPRS-R	**C**omputerized **P**atient **R**ecord **S**ystem **R**e-**E**ngineering
CPRS-RDV	**C**omputerized **P**atient **R**ecord **S**ystem **R**emote **D**ata **V**iews
CS	**C**ommon **S**ervices

DLL	**D**ynamic **L**ink **L**ibrary. A DLL allows executable routines to be stored separately as files with a DLL extension. These routines are only loaded when a program calls for them. DLLs provide several advantages: 1. DLLs help save on computer memory, since memory is only consumed when a DLL is loaded. They also save disk space. With static libraries, your application absorbs all the library code into your application so the size of your application is greater. Other applications using the same library will also carry this code around. With the DLL, you do not carry the code itself; you have a pointer to the common library. All applications using it will then share one image. 2. DLLs ease maintenance tasks. Because the DLL is a separate file, any modifications made to the DLL will not affect the operation of the calling program or any other DLL. 3. DLLs help avoid redundant routines. They provide generic functions that can be utilized by a variety of programs.
ENCRYPTION	Scrambling data or messages with a cipher or code so that they are unreadable without a secret key. In some cases encryption algorithms are one directional, that is, they only encode and the resulting data cannot be unscrambled (e.g., Access and Verify codes).
GUI	**G**raphical **U**ser **I**nterface
HOST	The term Host is used interchangeably with the term Server.
HRC	**H**ealth **R**evenue **C**enter
HSITES	**H**ealth **S**ystems **I**mplementation **T**raining and **E**nterprise **S**upport
LISTENER	The server process that listens for and accepts incoming connection requests from client applications.
MOVE!	Move! is a Web-based application and questionnaire for patient entry of weight management and physical activity with VistA. This is an interface via VistAWeb Web Services.
NON-CALLBACK RPC BROKER LISTENER	A server-side listener that does *not* use a separate port to call back to the RPCBroker in the GUI application (distributed and described with RPC Broker Patch XWB*1.1*35 and updated with XWB*1.1*44).
RPC	**R**emote **P**rocedure **C**all is a protocol that one program can use to request a service from a program located on another computer network. Essentially, M code can take optional parameters to do some work and then return either a single value or an array back to the client application.

SERVER	With respect to client/server software, server refers to the "receiving server" that sends the results in a message back to the "requesting server," where both servers reside in VistA on the same or on different VistA M systems. The server is where VistA M-based data and Business Rules reside, making these resources available to the requesting server. When the requesting server is receiving the results, it is referred to as the "server."
SIGN-ON/SECURITY	The Kernel module that regulates access to the menu system. It performs a number of checks to determine whether access can be permitted at a particular time. A log of signons is maintained.
TCPIP SERVICE	Multi-threaded listener implementation. The TCPIP multi-threaded service permits multiple TCPIP clients to connect and run as concurrent processes up to the limits established by the system. TCPIP listens on a particular port and launches the specified RPC Broker handler process for each client connection. The TCPIP is node-specific, so you *must* set up the TCPIP service for each node on which you want the service to run. Alos, the TCPIP service automatically creates log files in the RPCSERVER directory named VAH9200.LOG;xxx (where "xxx" is a file version number).

REF: For a comprehensive list of commonly used infrastructure- and security-related terms and definitions, please visit the Infrastructure & Security Services (ISS) Glossary Web page at the following Web address:

http://vaww.vista.med.va.gov/iss/glossary.asp

For a comprehensive list of acronyms, please visit the ISS Acronyms Web site at the following Web address:

http://vaww.vista.med.va.gov/iss/acronyms/index.asp

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S