HEALTHEVET WEB SERVICES CLIENT (HWSC)

INSTALLATION GUIDE

Version 1.0

February 2011

Revised May, 2013

Revision History

Table i. Revision History

Date	Description	Author(s)
05/2013	Reference to XOBT_1_0_Txx.zip. See page B-1.	HP VistA Maintenance Team. St Petersburg, Florida • David Elwell—Developer • Bob Sutton—Technical Writer.
02/2011	HWSC Version 1.0 Initial release	Product Development Services Security Program HWSC development team. Albany, NY OIFO: • Developer—Mike Kilmade • Developer—Liz Defibaugh Bay Pines, FL OIFO: • Development Manager—Charles Swartz Oakland, CA OIFO: • Developer—Kyle Clarke • Tester—Padma Subbaraman • SQA—Gurbir Singh • Technical Writer—Susan Strack

Contents

Contents

Tables

Figures

HealtheVet Web Services Client (HWSC)
Installation Guide
Version 1.0

1 Introduction

1.1 Document Overview

This document provides M-side installation and setup steps for the Health*e*Vet Web Services Client (HWSC) application. It is intended for M administrators at Veterans Affairs (VA) facilities, and it assumes familiarity with the following areas:

- Installing Kernel Installation and Distribution System (KIDS) file distributions on VistA/M servers

- VMS and Linux operating system management

Appendices A and B provide installation instructions for a sample web service application provided separately from the main HWSC release (that should only be installed in test accounts). The sample web service application has two parts:

- A WebLogic-based Web service (see Appendix A).

- An M client (see Appendix B) that uses HWSC to access the sample WebLogic service.

The appendices are provided primarily for M developers needing a sample to refer to when developing code that uses HWSC.

1.1.1   Additional Resources

The complete HWSC 1.0 end-user documentation package consists of:

- *HWSC 1.0 Installation Guide*

- *HWSC 1.0 System Management Guide*

- *HWSC 1.0 Developers Guide*

They are available from the Product Support anonymous directories and the VHA Software Documentation Library (VDL) Web site (http://www4.va.gov/vdl/application.asp?appid=180).

Health*e*Vet Web Services Client (HWSC) 1.0 end-user documentation and software can be downloaded from any of the **anonymous.software** directories on the Office of Information Field Office (OIFO) File Transfer Protocol (FTP) download sites:

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

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

- 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/

Health*e*Vet-VistA end-user 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/

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.

1.2 Known Caché and OpenVMS Issues Affecting HWSC

1.2.1   Problems with Mixed Caché Versions/Rolling Upgrades

Description: Several test sites for HWSC performed a "rolling" upgrade of some or all of their front end/commodity systems to Caché 2008.2.5 while temporarily leaving other systems including the database server at Caché v5.2.3. The result for PRE (an application using HWSC) was a predictable and repeatable communication failure for all communications using HWSC. The result for CPRS (an application using RDI and HWSC) included random user disconnections from CPRS, <PARAMETER> and <DISCONNECT> errors in the error trap, and occasional communications failures.

Workaround: None. Sites should upgrade all Caché systems within a configuration at the same time.

Affects: Any site that does a partial "rolling" upgrade, with some systems at one Caché version, and other systems at another Caché version.

Status: According to new guidance from InterSystems, sites should not mix Caché versions within a single environment when upgrading Caché versions. Affected sites addressed the issue by either upgrading all remaining systems to Caché 2008.2.5, or removing upgraded systems from the active configuration until all remaining systems could be upgraded.

1.2.2   <FUNCTION> Errors for RDI Outage Tasks

Description: When Remote Data Interoperability (RDI) v2 end-user processes detect a possible outage with the HDRII service, the RDI user process queues a background task to check the outage condition (RDI uses HWSC to communicate with HDRII.) In at least one case where two HDRII servers were in a state of FAILED, each RDI outage task failed with an error as follows:

$ZE= <FUNCTION>zSend+2^%Net.HttpRequest.1

The errors did not accumulate at a huge rate, and the failed task does not attempt to re-queue itself. The error itself does not pose a problem as determined by InterSystems. There is no impact on remote order checking as the error only occurs in the outage task, not in end-user processes. Internally, the error occurs in code that is looking for the current device.

Workaround: None, other than adding a screen to the error trap for <FUNCTION> errors. The errors stop once the HDRII outage is resolved.

Affects: Caché v2008.2.5 adhoc 9526 systems.

Status: At the time of HWSC 1.0 release, how to address the issue is being examined by InterSystems, the EIE HealtheVet Platform team, and the HWSC and RDI teams.

1.2.3   CACHETEMP Database Degrade Fix, Caché v5.2.3 adhoc 8574

Description: All Caché configurations include a CACHETEMP database/namespace, optimized for the storage of temporary data. During QA testing over several months on one specific Caché system, the shared CACHETEMP database/namespace became degraded on a number of occasions. This condition causes applications that rely on the ^CacheTemp global (including web service clients, as well as many InterSystems utilities) to fail. InterSystems determined the degraded ^CacheTemp global occurs due to an issue in Caché's internal exception handling (fixed by InterSystems internal patch number SAP1128). The relatively small buffer pool size on the particular test system may have exacerbated the issue, as well as the particular application usage patterns on the server, including XML parsing of SOAP and REST web service results using the Caché XML parser.

Workaround: None. HWSC 1.0- and applications using it should only be deployed in production on the Caché version which contains a fix for this issue (see below).

Affects: Many Caché versions, including VA-deployed versions v5.0.21 adhoc 6408 and v5.2.3 adhoc 5059.

Status: Fixed in:

- Caché v5.2.3 adhoc 8574 or higher
- Caché v2008.2.5 adhoc 9526 or higher
- all Caché v2008.2.6 versions
- all Caché v2009 and v2010 versions

1.2.4   Caché Objects XML Import on OpenVMS Systems

Description: When importing Caché Objects from an XML file on OpenVMS systems, the file must employ Windows, rather than OpenVMS, line break conventions.

Workaround: Place Caché Objects XML export files on OpenVMS file system, by transferring from a Windows system to the OpenVMS system in BINARY ftp mode.

Affects: Caché versions including v5.0.21 (VA Adhoc 6408) and v5.2.3 (VA Adhoc 5059).

Status: Appears to be fixed in Cache 2008.2.5 adhoc 9526.

1.2.5   %SOAP.WebClient Incorrect Content-Type

Description: On Caché v5.2.3 adhoc 5059, when making SOAP requests, an incorrect content-type (text/html) is assigned to the request (should be text/xml). Some SOAP web services reject requests with an incorrect content-type.

Workaround: Recommend applications use REST rather than SOAP style web service calls until all Caché deployment targets are upgraded to a version higher than v5.2.3 Adhoc 5059.

Affects: SOAP requests, in Caché v5.2.3 (VA Adhoc 5059) only. Does not affect REST requests.

Status: Fixed in Caché version 5.2.3 adhoc 8574.

1.2.6   OpenVMS SSL Library Memory Leak

Description: When calling SSL-protected web services (SOAP or REST) repetitively in the same M partition on OpenVMS systems, a memory leak occurs that eventually consumes all available memory, resulting in web service calls failing, and that in some cases can also result in a system freeze.

Workaround: None. Applications should not use SSL directly from Caché to connect to web services on OpenVMS systems until an OpenVMS fix is available, or until all Caché deployment targets have been migrated from VMS to another operating system (e.g., Linux). If it is absolutely required to secure web services with SSL, one other possibility is for Caché sites to employ a proxy server (a supported feature of Caché's SOAP and HTTP clients) to handle connections to SSL-protected web services rather than contacting such services directly from Caché.

Affects: All Caché versions running on OpenVMS.

Status: Awaiting a fix from HP (VMS systems). Would also be resolved by an operating system migration, e.g., a switch to Linux.

2 Installing and Configuring HWSC

2.1 Pre-Installation Information

2.1.1   Software Prerequisites (M-side)

- VistA Environment.

- Kernel, fully patched.

- **Caché version 2008.2.5 adhoc 9526 (or higher):** Installations on earlier Caché versions are not supported, and may result in installation or system failures.

- All systems in the configuration at the same Caché version.

> **WARNING: All systems within a configuration should be at the <u>SAME</u> Caché version when using Caché Objects-based applications such as HWSC. Mixed versions of Caché cause problems because compiled objects contain version-specific optimizations. Undefined, inconsistent application behavior results from Caché configurations with mixed versions, e.g., a database server running one version of Caché and an application server running another.**

2.1.2   Privileges/Staff Needed to Install

The installation of HWSC 1.0 requires both Mumps and Operating System privileges. Besides installation of a KIDS build, the install also requires some modifications at the operating system level, to adjust protection on several directories and one file.

> **At some facilities, operating system (e.g., VMS) and Mumps/VistA installation functions are performed by two different staffs. At such locations, two installer personnel may be required, one from each staff group.**
>
> **Also, since installs may need to be re-run more than once if an error is encountered, personnel from both staff groups should be available for the duration of the installation, until a successful install has been achieved.**

The following privileges/permissions are necessary to perform the installation:

1. **Operating system (OS) system administrator privileges**: To perform some of the Pre-Installation Steps and Post-Installation Steps.

2. **Privileged Cache account access:** To perform some of the Pre-Installation Steps, the Caché process will need rights greater than the %developer role, for example the %All role.

3. **Programmer access:** DUZ(0)="@" is required for installing the HWSC 1.0 KIDS build.

2.1.3   XOBW Distribution ZIP File Structure

(root)
|-- readme.txt (last minute changes)
|-- XOBW_1_0_Bxx.KID (KIDS build)
|-- XOBW_1_0_Bxx.XML (Caché Objects Import File, where "xx" is the build number)
|-- XOBW_1_0_Bxx.XML.MD5 (MD5 checksum for corresponding XML file)

i   **NOTE:** There is no required Java/J2EE component for the core HWSC (XOBW) application.

2.1.4   Estimated Installation Time

The estimated installation time for XOBW is less than two minutes.

2.1.5   System Processes

- VistALink users can remain on the system.
- Roll-and-scroll and RPC Broker users can remain on the system.
- TaskMan does not need to be put into a wait state.

2.1.6   Routine/Global Namespaces

HWSC 1.0 has been assigned the XOBW routine/global namespace.

2.1.7   File and Global Information

HWSC 1.0 installs the M files shown in the following table.

Table 2-1. HWSC M Files

File #	File Name	Root Global	FileMan Protection
18.02	WEB SERVICE	^XOB(18.02,	@
18.12	WEB SERVER	^XOB(18.12,	@
18.13	WEB SERVICE LOOKUP KEY	^XOB(18.13,	@

These files, at installation, consume less than 2000 bytes of space. Entries added to these files will be small, and the quantity of entries added will also be small. The space used will increase only by the size of added entries.

2.1.8 Journaling

Current VA recommendations are that journaling should be enabled for all globals, and should not be turned off. The ^XOB global is relatively static, and its journaling will result in very little overhead.

2.1.9 Global Protection

The XOB global should be protected such that user and TaskMan processes are granted RW access.

On Caché 2008.2.5 systems, the ^XOB global should be protected with a resource that grants RW privileges, e.g.:

%DB_%DEFAULT:RW

Note: For previous Mumps versions, this level of protection had typically been described as:

Table 2-2. M System Protection

Global Name	Caché	
^XOB	Owner:	RWD
	Group:	N
	World:	N
	Network:	RWD

2.1.10 Global Placement, Mapping, and Translation

HWSC utilizes one global, ^XOB. For virgin installs, ^XOB should be placed in a location appropriate for a small, static global, prior to installation of the HWSC application in Caché. For M configurations with multiple databases or volume sets, any necessary mapping or translation should be set up at this time as well.

2.1.11 Routine Checksums

The routine name and corresponding checksum value for each M routine contained within the HWSC 1.0 software package are listed below.

Table 2-3. HWSC Routine Checksums

Routine	Checksum
XOBWD	38463620
XOBWENV	7338181
XOBWLIB	23307912
XOBWLIB1	32808293
XOBWPST	17271805
XOBWPWD	11849037
XOBWSSL	17552025
XOBWU	8679055
XOBWU1	20116927
XOBWUA	14787845
XOBWUA1	14207560
XOBWUS	6230795
XOBWUS1	56242034
XOBWUS2	749555

i **NOTE:** Checksums were created using CHECK1^XTSUMBLD.

2.2 Pre-Installation Steps (All Caché Versions)

i **NOTE:** Some of the instructions that follow must be performed according to the target Operating System and may need to be performed on each instance of your Caché configuration, including configurations with a mixed-OS environment; for example, VMS/Linux or VMS/Windows.

2.2.1 Ensure Adequate VMS Process Parameters (Quotas) for All End-User and TaskMan Accounts

Requires System Administrator Privileges

On VMS systems, ensure that the VMS accounts used for end-user processes and TaskMan tasks have adequate VMS process parameters (quotas). The following table lists the recommended minimum values from InterSystems for certain process parameters.

Table 2-4. Recommended VMS Process Parameters (Quotas)

VMS Process Parameter/Quota	InterSystems Recommended Minimum
ASTLM	300
BIOLM	300
BYTLM	300,000
DIOLM	300
FILLM	255
PGFLQUOTA	300,000
TQELM	300
WSQUOTA	2048/3192
WSEXTENT	8192/16384
ENQLM	3000

If VMS process parameters are not set to at least the minimum values recommended by InterSystems, calls to external web services made in a given end-user's process may fail. Errors messages may include the phrase:

```
"Error: <FUNCTION>zDelete^%ooLibrary.File.1"
```

For more information and the current recommended minimum values for the VMS process parameters, see http://docs.intersystems.com/csp/docbook/GCI_vmsparmcalc.html.

2.2.2 Verify or Set ^%SYS("TempDir") On All Caché Instances

ⓘ **NOTE:** ^%SYS("TempDir") is already configured on most if not all VA production systems. Setting it is also a part of the ZSTU startup routine, which sets it based on the value of Kernel's DEFAULT DIRECTORY FOR HFS site parameter. Therefore, on VA production systems you will likely find it is already set appropriately on each Caché instance/node in your configuration.

Requires: Privileged Cache account access (higher than the %developer role, for example the %All role)

HWSC uses Caché classes to consume web services. These classes require the use a host file system directory to create, read, write and delete temporary files. The directory to use must be configured for Caché, and then set up to ensure RWED access for all the processes that might use it.

Setting ^%SYS("TempDir") on each system in your configuration, is the recommended way to configure/direct the Caché classes to use a specific directory for temporary files.

We recommend using the same directory that is used as the DEFAULT DIRECTORY FOR HFS field configured in the Kernel Site Parameters option, This corresponds to the PRIMARY HFS DIRECTORY field in the Kernel System Parameters file (#8989.3). In particular, assuming a directory is provided in this field, it should already be set up for RWED access for Caché processes. Very recent versions of the ^ZSTU startup routine set this setting/location automatically.

The ^%SYS("TempDir") settings need to be **permanent**; they are not just for the installation.

On mixed OS configurations, the directory location has to be specific to the OS (e.g., a VMS temporary directory on VMS systems and a Linux temporary directory on Linux systems). Also, on mixed-OS configurations, the SECONDARY HFS DIRECTORY is defined in the Kernel Site Parameters option to show the HFS directory in the secondary non-VMS OS (Linux or Windows).

NOTE: ^%SYS is not mapped across systems, and therefore needs to be individually set on each instance/box/system.

To **check**, and if necessary **set** ^%SYS("TempDir") on each instance/box/system:

1. On every system/box/instance in your configuration, first check if ^%SYS("TempDir") is defined. If so, verify that it pointing to a directory that has W:RWED permissions. If so, skip to step 5 below.

2. If ^%SYS("TempDir") is not set, or if it is set to a directory that does not exist or does not have W:RWED permissions, continue.

   Determine the value of the Kernel System Parameter DEFAULT DIRECTORY FOR HFS on your system. For convenience this value can also be obtained from the extrinsic function $$DEFDIR^%ZISH(). E.g.,

   ```
   W $$DEFDIR^%ZISH()
   ```

   On mixed-OS configurations, this extrinsic function will return the value of the PRIMARY HFS DIRECTORY when the OS is VMS; and will return the value of the SECONDARY HFS DIRECTORY when the OS is non-VMS.

3. Verify that this directory has RWED access for the user processes that access Caché.

4. If it is not set, set ^%SYS("TempDir") to this directory location. E.g.,

   ```
   S ^%SYS("TempDir")="USER$:[TEMP]"
   ```

   (replace USER$:[TEMP] with a value appropriate for your operating system).

   Or for convenience, you can set it with the extrinsic function. E.g.,

```
S ^%SYS("TempDir")=$$DEFDIR^%ZISH()
```

5. If you wish to test the **^%SYS("TempDir")** setting, refer to Appendix D.

6. Repeat steps 1-5 on all the Caché boxes/instances in your configuration.

```
S ^%SYS("TempDir")=$$DEFDIR^%ZISH()
```

2.3 Installation Steps

Requires: Programmer access (DUZ(0)="@")

i **NOTE:** The Pre-Installations Steps must be performed first to ensure that the temporary directories being used during this step have the adequate access.

2.3.1   Place XOBW Installation Files on M Server File System

On configurations with a back-end database server and front-end application servers/commodity boxes, <u>log onto the database server</u> and place the XOBW installation files on that system's file system.

2.3.1.1   Protection Settings for M Server Folder Containing XOBW Installation Files

On OpenVMS and Linux systems, the folder you place the XOBW installation files in should allow at least W:RE (world read/execute) access.

2.3.1.2   Transfer XOBW Installation Files to M Server File System

The two files to transfer to the M Server file system are:

- XOBW_1_0_Bxx.KID (KIDS build)
- XOBW_1_0_Bxx.XML (Cache Objects export file)

Transfer both XOBW installation files to the M server using normal mechanisms, e.g., ASCII mode ftp.

2.3.2   Load and Install Distribution

On configurations with a back-end database server and front-end application servers/commodity boxes, <u>log onto the database server</u> and perform the KIDS installation on that system.

To install HWSC, load the XOBW KIDS distribution from the distribution file (XOBW_1_0_Bxx.KID), perform other KIDS steps (e.g., Verify Checksums) as necessary, and then install the loaded XOBW KIDS distribution.

i **NOTE:** Programmer access (DUZ(0)="@") is required for installing the KIDS build.

2.3.3   Enter Directory for XOBW_1_0_Bxx.xml File

During the KIDS install for HWSC, when you see the install prompt "`For XOBW xml install file, enter directory location:`" enter the directory location (on the host file system of the Caché system) where you have placed the XOBW_1_0_Bxx.xml file. The file will be imported as part of the post-init. The default directory location presented is the default defined by the Kernel Site Parameters.

If applicable for the operating system, the directory name entered should contain the operating system-specific delimiter as the last character:

- Linux Example: /home/cache/   where '/' is the delimiter
- Windows Example: c:\tmp\cache\   where '\' is the delimiter

2.3.4   Sample KIDS Installation

Figure 2-1: Example KIDS Installation: HWSC (M-Side)

```
Select Installation Option: 1 <Enter> Load a Distribution

Enter a Host File: VA4$:[ANONYMOUS.ANONYMOUS.HWSC]XOBW_1_0_B31.KID

KIDS Distribution saved on Jun 27, 2007@05:50:02
Comment: XOBW release candidate build 31

This Distribution contains Transport Globals for the following Package(s):
   XOBW 1.0
Distribution OK!

Want to Continue with Load? YES// y <Enter> YES
Loading Distribution...

Build XOBW 1.0 has an Environmental Check Routine
Want to RUN the Environment Check Routine? YES// <Enter>
   XOBW 1.0
Will first run the Environment Check Routine, XOBWENV

 >>> Environment check completed for KIDS Load a Distribution option.

Use INSTALL NAME: XOBW 1.0 to install this Distribution.

   1      Load a Distribution
   2      Verify Checksums in Transport Global
   3      Print Transport Global
   4      Compare Transport Global to Current System
   5      Backup a Transport Global
   6      Install Package(s)
          Restart Install of Package(s)
          Unload a Distribution
```

```
Select Installation Option: 6 <Enter> Install Package(s)
Select INSTALL NAME: XOBW 1.0 <Enter> Loaded from Distribution    Loaded from
Distribution  6/27/07@11:03:24
     => XOBW release candidate build 31  ;Created on Jun 27, 2007@05:50:02

This Distribution was loaded on Jun 27, 2007@11:03:24 with header of
   XOBW release candidate build 31  ;Created on Jun 27, 2007@05:50:02
   It consisted of the following Install(s):
       XOBW 1.0
Checking Install for Package XOBW 1.0
Will first run the Environment Check Routine, XOBWENV

 >>> Environment check completed for KIDS Install Package option.

Install Questions for XOBW 1.0

Incoming Files:

   18.02     WEB SERVICE

   18.12     WEB SERVER

   18.13     WEB SERVER LOOKUP KEY

Want KIDS to Rebuild Menu Trees Upon Completion of Install? YES// n <Enter> NO

For XOBW xml install file, enter directory location: USER$:[TEMP]

Want KIDS to INHIBIT LOGONs during the install? YES// n <Enter> NO
Want to DISABLE Scheduled Options, Menu Options, and Protocols? YES// n <Enter> NO

Enter the Device you want to print the Install messages.
You can queue the install by enter a 'Q' at the device prompt.
Enter a '^' to abort the install.

DEVICE: HOME// <Enter>  TERMINAL

 Install Started for XOBW 1.0 :
               Jun 27, 2007@11:04:43

Build Distribution Date: Jun 27, 2007

 Installing Routines
               Jun 27, 2007@11:04:43

 Installing Data Dictionaries:
               Jun 27, 2007@11:04:43

 Installing PACKAGE COMPONENTS:

 Installing INPUT TEMPLATE

 Installing DIALOG

 Installing PROTOCOL
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
  Located in the XOBW (WEB SERVICES CLIENT) namespace.
```

```
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.
   Located in the XOBW (WEB SERVICES CLIENT) namespace.

 Installing LIST TEMPLATE

 Installing OPTION
                Jun 27, 2007@11:04:44

 Running Post-Install Routine: EN^XOBWPST

 o  Deleting xobw classes:

       ...[xobw] deletion finished successfully.

Load started on 06/27/2007 11:04:47
Loading file USER$:[TEMP]XOBW_1_0_B31.XML as xml
Imported class: xobw.RestRequest
Imported class: xobw.RestRequestFactory
Imported class: xobw.VistaInfoHeader
Imported class: xobw.WebServer
Imported class: xobw.WebServiceMetadata
Imported class: xobw.WebServiceProxyFactory
Imported class: xobw.WebServicesAuthorized
Imported class: xobw.WsdlHandler
Imported class: xobw.error.AbstractError
Imported class: xobw.error.BasicError
Imported class: xobw.error.DialogError
Imported class: xobw.error.HttpError
Imported class: xobw.error.ObjectError
Imported class: xobw.error.SoapError
Compiling class xobw.RestRequest
Compiling class xobw.RestRequestFactory
Compiling class xobw.VistaInfoHeader
Compiling class xobw.WebServer
Compiling class xobw.WebServiceMetadata
Compiling class xobw.WebServiceProxyFactory
Compiling class xobw.WebServicesAuthorized
Compiling class xobw.WsdlHandler
Compiling class xobw.error.AbstractError
Compiling class xobw.error.BasicError
Compiling class xobw.error.DialogError
Compiling class xobw.error.HttpError
Compiling class xobw.error.ObjectError
Compiling class xobw.error.SoapError
Compiling table xobw.WebServer
Compiling table xobw.WebServiceMetadata
Compiling table xobw.WebServicesAuthorized
Compiling routine xobw.RestRequest.1
Compiling routine xobw.RestRequestFactory.1
Compiling routine xobw.VistaInfoHeader.1
Compiling routine xobw.WebServer.1
Compiling routine xobw.WebServiceMetadata.1
Compiling routine xobw.WebServiceProxyFactory.1
Compiling routine xobw.WebServicesAuthorized.1
```

```
Compiling routine xobw.WebServicesAuthorized.2
Compiling routine xobw.WsdlHandler.1
Compiling routine xobw.error.AbstractError.1
Compiling routine xobw.error.BasicError.1
Compiling routine xobw.error.DialogError.1
Compiling routine xobw.error.HttpError.1
Compiling routine xobw.error.ObjectError.1
Compiling routine xobw.error.SoapError.1
Load finished successfully.

   Support classes imported successfully.

 Updating Routine file...

 Updating KIDS files...

 XOBW 1.0 Installed.
               Jun 27, 2007@11:05

 Install Message sent #2

Install Completed
```

2.4 Post-Installation Steps

2.4.1   Troubleshoot Installation Errors/Review Install File

Review the contents of the install file and verify that no errors occurred. If an error did occur, check the following troubleshooting items to see if any match the error encountered.

If installation of HWSC 1.0 fails, the recommended action is to review the install logs, determine and address the cause of install failure, and then re-run the HWSC installation. The main reason for HWSC installation failures is file protection settings on VMS systems. The HWSC installation can be re-run as many times as necessary until a successful installation is achieved.

Some common installation errors are listed below.

2.4.1.1   Caché Error 6301 SAX XML Parser

On an OpenVMS system, if during installation, a SAX XML parser error #6301 like the following is listed during the import of the Caché Objects support classes file, with a reference to "<MAXSTRING>error+6":

Figure 2-2 Cache Error 6301 SAX XML Parser: <MAXSTRING>error+6

```
  Directory: USER$:[TEMP]
  File Name: XOBW_1_0_B31.XML
     Error: ERROR #6301: SAX XML Parser Error: <MAXSTRING>error+6
o  Classes not imported.
```

Alternatively, you may encounter a series of Caché 6301 errors like the following, specifically referring to "cacheexport.xsd":

Figure 2-3 Cache Error 6301 SAX XML Parser: Primary Document Could Not Be Opened

```
Error: ERROR #6301: SAX XML Parser Error: Line: 2 Offset: 125 An exception
 occurred! Type:RuntimeException, Message:Warning: The primary document entity
could not be opened. Id=_$1$DISK:[CACHESYS.ISC1.BIN]cacheexport.xsd at line 0
offset 0
```

And/or, if a series of errors (referring to undeclared attributes and unknown elements) is displayed following that error, similar to the following:

Figure 2-4 Undeclared Attributes and Unknown Elements

```
Line: 2 Offset: 125 An exception occurred! Type:RuntimeException, Message:Warning:
The primary document entity could not be opened.
Id=$1$DISK:[CACHESYS.ISC1.BIN]cacheexport.xsd at line 0 offset 0Line: 2 Offset: 125
Unknown element 'Export' while processing
$1$DISK:[CACHESYS.ISC1.MGR.TEMP]568337052XWMf1.XML at line 2 offset 125
Line: 2 Offset: 125 Attribute 'generator' is not declared for element 'Export'
while processing $1$DISK:[CACHESYS.ISC1.MGR.TEMP]568337052XWMf1.XML at line 2
offset 125
Line: 2 Offset: 125 Attribute 'version' is not declared for element 'Export' while
processing $1$DISK:[CACHESYS.ISC1.MGR.TEMP]568337052XWMf1.XML at line 2 offset 125
Line: 2 Offset: 125 Attribute 'zv' is not declared for element 'Export' while
processing $1$DISK:[CACHESYS.ISC1.MGR.TEMP]568337052XWMf1.XML at line 2 offset 125
Line: 2 Offset: 125 Attribute 'ts' is not declared for element 'Export' while
processing $1$DISK:[CACHESYS.ISC1.MGR.TEMP]568337052XWMf1.XML at line 2 offset 125
Line: 3 Offset: 32 Unknown element 'Class' while processing
$1$DISK:[CACHESYS.ISC1.MGR.TEMP]568337052XWMf1.XML at line 3 offset 32
```

To fix, access must be opened up to the file cacheexport.xsd, as well as the directory containing it.

1. Navigate to the Caché install directory location for this Caché system.
2. Locate the file cacheexport.xsd, typically in the [BIN] subdirectory.
3. Open up W:RE (world: read execute) access to the directory containing cacheexport.xsd (in this example, 1DISK:[CACHESYS.ISC1.BIN].
4. Open up W:RE access to the cacheexport.xsd file itself.
5. Re-run the HWSC KIDS install.

2.4.1.2 Caché Error 6301 SAX XML Parser: Invalid Document Structure

On an OpenVMS system, if during installation, a SAX XML parser error #6301 with 'Invalid Document Structure', like the following is listed during the import of the Caché Objects support classes file, with a reference to invalid document structure at line 1 offset 1:

Figure 2-5. Error 6301 SAX XML Parser: Invalid Document Structure

```
Error occurred during the importing of support classes file:
  Directory: $1$DGA2:[CLARKE.HOME]
```

```
   File Name: XOBW_1_0_B31.XML
      Error: ERROR #6301: SAX XML Parser Error: Invalid document structure while
processing _$1$DGA2:[CACHESYS.ISCISC6A1.MGR.TEMP]539532000UHNo1.XML at line 1
offset 1
   Classes not imported.
```

On v5.2.3 Caché systems on OpenVMS only, there was a bug where the XML host file containing the Caché Objects classes needed to be ftp's in BINARY rather than ASCII mode, otherwise this error would occur. The fix was to re-transmit the XML file in BINARY mode and re-run the installation. This error should not be encountered on Caché v2008.2.5 systems.

2.4.1.3   Caché Error 5024 Unable to Copy File

If, during installation, an error #5024 like the following is listed during the import of Caché Objects support classes file:

Figure 2-6Cache Error 5024: Unable to copy file

```
Error occurred during the importing of support classes file:
  Directory: ABC_HFS$:[TMP]
  File Name: XOBW_1_0_B31.XML
      Error: ERROR #5024: Unable to copy file '_DSA157:[ABC.TMP]XOBW_1_0_B31.XML
' to '_DSA158:[CACHESYS.ABC1TSVR.MGR.TEMP]592122798aYuW1.XML'
 o  Classes not imported.
```

To fix, RWED access must temporarily be opened up to TEMP directory noted in "copy to" destination in the error message. Then re-run the HWSC KIDS install.

2.4.1.4   Bad Global Mapping ("Error with Unknown status code")

If an error is listed like the following during the import of the Caché Objects support classes file, with "Error #0: Unknown status code: 6301", it may be a bad global mapping.

Figure 2-7. Error #0: Unknown status code: 6301

```
Error occurred during the importing of support classes file:
  Directory: SYS$SYSDEVICE:[ANONYMOUS]
  File Name: XOBW_1_0_B31.XML
      Error: ERROR #0: Unknown status code: 6301
 o  Classes not imported.
```

Some Caché accounts may have a global mapping entry that maps %* to %SYS. In most cases this entry should be removed; at the time of release of HWSC, it is no longer part of the VA cookbook recommendations for global mapping, and it will interfere with the proper mapping of % globals used by Caché Object classes.

2.4.1.5   Caché Error 5075 (Class Dictionary Out of Date)

This very rare error should never be encountered on production VA systems. If encountered, it would occur during installation of the KIDS file in Caché, if the Caché class dictionaries in the account you're installing in are not up-to-date:

ERROR #5075: Class dictionary out of date, please run upgrade utility $system.OBJ.Upgrade()

This error typically means that when the cache.dat file for the account was upgraded for the new version of Caché, a required step to upgrade the Caché Object classes (for example, in the Caché v5 upgrade) in the namespace was not performed. Because Caché Objects are now being used in certain VistA applications (including HWSC), recent Caché upgrade instruction documents used by VA now include steps to recompile Caché Objects.

To fix this issue if encountered during HWSC installation:

- VA production systems: Contact the VA IT Engineering HSTS team. You will need to revisit part of the Caché upgrade process for your system, specifically the section involving recompiling objects.

- VA non-production systems (e.g., developer or QA accounts): Contact the VA IT Engineering HSTS team or obtain the Caché upgrade instructions document from that team's web page ('AXP Alert Messages'), for your Caché version and operating system. You will need to revisit the part of those instructions that pertains to recompiling objects, involving the following commands:

    o $SYSTEM.OBJ.Upgrade()
    o $SYSTEM.OBJ.CompileAll()

- Non-VA systems: Refer to InterSystems' upgrade instructions for your Caché version (including operating system), in particular the section of the install guide that involves recompiling objects.

2.4.2   Assign XOBW WEB SERVER MANAGER Option in Menu Manager

The XOBW WEB SERVER MANAGER option is the management user interface for HWSC.

This option should be restricted, and assigned to personnel in Information Resources Management (IRM) – or equivalent – organization only. Add it to the appropriate menu option(s) accessible by IRM personnel only.

Appendix A: Installation Back-Out/Roll-Back Procedure

HWSC 1.0 is a brand-new application. As such there is no back-out/roll-back procedure to revert to a previous version.

If installation of HWSC 1.0 fails, the recommended action is to review the install logs, determine and address the cause of install failure, and then re-run the HWSC installation. The main reason for HWSC installation failures is file protection settings on VMS systems. The HWSC installation can be re-run as many times as necessary until a successful installation is achieved.

Appendix B: Installing the "Test" Sample Java Web Service

This appendix provides supplementary installation and setup steps for a sample "tester" application, provided in a separate, non-production, unreleased XOBT-namespaced distribution zip. These installation instructions assume familiarity with WebLogic systems administration.

For Non-Production Systems Only

The sample java web service is intended for installation on non-production systems only (e.g., developer systems). It is a sample application to demonstrate the use of HWSC to developers.

XOBT_1_0_T.xx Zip File

The un-released, separately distributed XOBT_1_0_Txx.zip distribution file contains the sample application (M client and java web service sides).

> **REF:** For more information on how to obtain the XOBT_1_0_Txx.zip distribution file, contact the OIT PD VistA Maintenance HWSC Developers (Outlook Mail Group: OITPDVistAMaintenanceHWSCDevelopers@va.gov)..

```
 (root)
|-- readme.txt (last minute changes)
|
|-- sample-ears (EAR files for sample web service)
|    |-- wls9x (version for WebLogic 9.x/10.x)
|    |-- wls8.1 (version for WebLogic 8.1)
|
|-- sample-m (M-based sample web service client application)
|    |-- XOBT_1_0_Txx.KID (KIDS build)
|    |-- HwscTesterWebService.wsdl (WSDL file to import for sample web service)
|
|-- sample-prj (sample web service source files with ANT build file)
```

Overview

The EAR file `hwscSample-1.0.0.xxx.ear`, in the separate, non-production, unreleased XOBT_1_0_Txx.zip, contains two pre-built "tester" sample web services you can deploy in WebLogic 8.1:

- SOAP-style web service (`hwscSoapSampleWs.war`)
- REST-style web service (`hwscRestSampleSvc.war`)

The sample web services provide "out of the box" functionality that can be called by the corresponding sample M sample client code, also distributed with HWSC (see Appendix B). The tester Java web service enables developers to do the following from the M sample client code:

- Confirm the connection with a server
- Retrieve dummy data from a server
- Enter text and echo it back from a server
- Retrieve system properties from a server.

NOTE: SOAP-based web service functionality is supported in the "tester" web service via the third-party open source Java SOAP framework XFire. All the files required for the XFire framework are contained in `hwscSoapSampleWs.war`. There is no need to download XFire.

Prerequisites

- BEA WebLogic 8.1 (SP4 or greater) or BEA WebLogic 9.x/10.x
- JVM 1.4.2+

Tester Service Installation Steps

Add User Group

Add a security group called "XOBW_Server_Proxies" to the default WebLogic security realm.

Add User as Group Member

Assign at least one WebLogic user to the XOBW_Server_Proxies group

(WebLogic 9.x/10.x) Add a Global Role

Add a Global Security Role:

- Name: XOBW_Server_Proxies
- Condition: Group: XOBW_Server_Proxies

Deploy Tester Web Service

Deploy **hwscSample-1.0.0.xxx.ear** via the WebLogic console in the usual manner. Use the EAR built for the appropriate WebLogic version. The EAR files are located in the **sample-ears** folder of the XOBT zip distribution (subfolders: **wls81** for WebLogic 8.1, and **wls9.x** for WebLogic 9.x/10.x).

Appendix C: Installing the "Test" Sample Application (M-Side)

This appendix provides supplementary installation and setup steps in a separate, non-production, unreleased XOBT-namespaced distribution zip. This is intended primarily for M developers needing a sample to refer to when developing code that uses HWSC. These installation instructions assume familiarity with the installation of KIDS file distributions on VistA/M servers.

For Non-Production Systems Only

The M-side "Test" sample application is intended for installation on non-production systems only (e.g., developer systems). It is a sample application to demonstrate the use of HWSC to developers.

Overview

The M-side "Test" sample application is used to demonstrate consuming a J2EE-based web service from Caché. It is provided to demonstrate:

- basic coding of web service clients (for development purposes)
- basic HWSC connectivity and management (for system administrators).

The corresponding web service it consumes is also distributed by HWSC; to install the sample web service on the Java side, see *Appendix A: Installing the "Tester" Sample Java Web Service*.

The un-released, separately distributed XOBT_1_0_Txx.zip distribution file contains the sample application (M client and java web service sides).

Pre-Installation Information

Prerequisites (M-side)

- VistA Environment.
- Kernel, fully patched.
- **Caché version 2008.2.5 adhoc 9526 (or higher):** Installations on earlier Caché versions are not supported, and may result in installation or system failures.
- All systems in the configuration at the same Caché version.
- Health*e*Vet Web Services Client (HWSC) 1.0

Estimated Installation Time

The estimated installation time for XOBT is less than 2 minutes.

System Processes

- VistALink users can remain on the system
- Roll-and-scroll and RPC Broker users can remain on the system
- TaskMan does not need to be put into a wait state.

M Server Permissions

Programmer access (DUZ(0)="@") is required for installing the KIDS build.

Namespaces

The HWSC "test" sample application has been assigned the XOBT namespace.

File and Global Information

The HWSC "test" sample application does not export any files.

Journaling

n/a

Protection

n/a

Global Placement, Mapping and Translation

n/a

Checksum Information

The routine name and corresponding checksum value for each M routine contained within the HWSC 1.0 software package are listed below.

Table C-1. XOBT (Sample Tester) Routine Checksums

Routine	Checksum
XOBTENV	3765263
XOBTPST	19585247
XOBTWRA	37758194
XOBTWRA1	15475590
XOBTWRB	29772296
XOBTWRB1	25075453
XOBTWSA	77013589
XOBTWSA1	50571990
XOBTWSB	38222979
XOBTWSB1	54224405
XOBTWU	18433100
XOBTWU1	34288738

i **NOTE:** Checksums were created using CHECK1^XTSUMBLD.

Using the Sample Application

For information on using the sample application, please see the *HWSC Developer Guide*.

Installation Steps

Place XOBT Installation Files on M Server File System

Protection Settings for M Server Folder Containing XOBT Installation Files

On OpenVMS and Linux systems, the folder you place the XOBT installation files in should allow at least W:RE (world read/execute) access.

Transfer Sample Installation Files to M Server File System

The two files to transfer to the M Server file system are:

- XOBT_1_0_Txx.KID (KIDS build)
- HwscTesterWebService.wsdl (Web Service description file)

Both of these files can be transferred in ASCII ftp mode on all operating systems.

Load and Install All Files

To load and install all files, simply run the KIDS install, **XOBT_1_0_Txx.KID.**

NOTE: Programmer access (DUZ(0)="@") is required for installing the KIDS build.

Enter Directory Location for Sample WSDL File

When you see the install prompt

```
For WSDL install file, enter directory location:
```

enter the directory location (on the host file system of the Caché system) where you have placed the **HwscTesterWebService.wsdl** file. The file will be imported as part of the post-init. The default directory location presented is the default defined by the Kernel Site Parameters.

If applicable for the operating system, directory name entered should contain the operating system specific delimiter as the last character:

- Linux Example: /home/cache/ where '/' is the delimiter
- Windows Example: c:\tmp\cache\ where '\' is the delimiter

Sample KIDS Installation

Figure C-1. Example KIDS Installation: "Test" Sample Application (M-Side)

```
Select Installation Option: 1 <Enter> Load a Distribution
Enter a Host File: C:\temp\XOBT_1_0_T31.KID

KIDS Distribution saved on Jun 27, 2007@05:50:35
Comment: XOBT release candidate build 31

This Distribution contains Transport Globals for the following Package(s):
   XOBT 1.0
Distribution OK!

Want to Continue with Load? YES// y <Enter> YES
Loading Distribution...

Build XOBT 1.0 has an Environmental Check Routine
Want to RUN the Environment Check Routine? YES// y <Enter> YES
   XOBT 1.0
Will first run the Environment Check Routine, XOBTENV
```

```
 >>> Environment check successfully completed for KIDS Load a Distribution option.

Use INSTALL NAME: XOBT 1.0 to install this Distribution.

    1         Load a Distribution
    2         Verify Checksums in Transport Global
    3         Print Transport Global
    4         Compare Transport Global to Current System
    5         Backup a Transport Global
    6         Install Package(s)
              Restart Install of Package(s)
              Unload a Distribution

Select Installation Option: 6 <Enter> Install Package(s)
Select INSTALL NAME: XOBT 1.0 <Enter> Loaded from Distribution     Loaded from
Distribution  6/27/07@11:10:10
      => XOBT release candidate build 31  ;Created on Jun 27, 2007@05:50:35

This Distribution was loaded on Jun 27, 2007@11:10:10 with header of
   XOBT release candidate build 29  ;Created on Jun 27, 2007@05:50:35
   It consisted of the following Install(s):
        XOBT 1.0
Checking Install for Package XOBT 1.0
Will first run the Environment Check Routine, XOBTENV


 >>> Environment check successfully completed for KIDS Install Package option.


Install Questions for XOBT 1.0

For WSDL install file, enter directory location: c:\temp\

Want KIDS to INHIBIT LOGONs during the install? YES// n <Enter> NO
Want to DISABLE Scheduled Options, Menu Options, and Protocols? YES// n <Enter> NO

Enter the Device you want to print the Install messages.
You can queue the install by enter a 'Q' at the device prompt.
Enter a '^' to abort the install.

DEVICE: HOME// <Enter> TERMINAL


 Install Started for XOBT 1.0 :
              Jun 27, 2007@11:11:18

Build Distribution Date: Jun 27, 2007

 Installing Routines
              Jun 27, 2007@11:11:18

 Installing PACKAGE COMPONENTS:

 Installing PROTOCOL
  Located in the XOBT (WEB SERVICES CLIENT TESTER) namespace.
  Located in the XOBT (WEB SERVICES CLIENT TESTER) namespace.
  Located in the XOBT (WEB SERVICES CLIENT TESTER) namespace.
  Located in the XOBT (WEB SERVICES CLIENT TESTER) namespace.
  Located in the XOBT (WEB SERVICES CLIENT TESTER) namespace.
  Located in the XOBT (WEB SERVICES CLIENT TESTER) namespace.

 Installing LIST TEMPLATE
```

```
                Jun 27, 2007@11:11:18

 Running Post-Install Routine: EN^XOBTPS

 o  Deleting xobt classes:

       ...[xobt.soap] deletion finished successfully.

       ...[xobt.rest] deletion finished successfully.

       ...[hwsc] deletion finished successfully.

       ...[xobt] deletion finished successfully.

Compilation started on 06/27/2007 11:11:24 with qualifiers 'dk'
Compiling class xobt.soap.EmployeeInformationVO
Compiling class xobt.soap.EmployeeList
Compiling class xobt.soap.GetEmployeeInformationFault
Compiling routine xobt.soap.EmployeeInformationVO.1
Compiling routine xobt.soap.EmployeeList.1
Compiling routine xobt.soap.GetEmployeeInformationFault.1
Compilation finished successfully.

Compilation started on 06/27/2007 11:11:24 with qualifiers 'dk'
Compiling class xobt.soap.TesterWebServiceHttpPort
Compiling routine xobt.soap.TesterWebServiceHttpPort.1
Compiling class xobt.soap.TesterWebServiceHttpPort.doArrayInput
Compiling class xobt.soap.TesterWebServiceHttpPort.doBoolean
Compiling class xobt.soap.TesterWebServiceHttpPort.doCalendar
Compiling class xobt.soap.TesterWebServiceHttpPort.doDate
Compiling class xobt.soap.TesterWebServiceHttpPort.doDouble
Compiling class xobt.soap.TesterWebServiceHttpPort.doEcho
Compiling class xobt.soap.TesterWebServiceHttpPort.doEchoEmployeeInfoVO
Compiling class xobt.soap.TesterWebServiceHttpPort.doEchoXml
Compiling class xobt.soap.TesterWebServiceHttpPort.doFloat
Compiling class xobt.soap.TesterWebServiceHttpPort.doInt
Compiling class xobt.soap.TesterWebServiceHttpPort.doLong
Compiling class xobt.soap.TesterWebServiceHttpPort.doPing
Compiling class xobt.soap.TesterWebServiceHttpPort.doShort
Compiling class xobt.soap.TesterWebServiceHttpPort.doSleep
Compiling class xobt.soap.TesterWebServiceHttpPort.getDOI
Compiling class xobt.soap.TesterWebServiceHttpPort.getEmployeeInfo
Compiling class xobt.soap.TesterWebServiceHttpPort.getEmployeeInfoVO
Compiling class xobt.soap.TesterWebServiceHttpPort.getEmployeeList
Compiling class xobt.soap.TesterWebServiceHttpPort.getEmployeeListOnly
Compiling class xobt.soap.TesterWebServiceHttpPort.getGettysburgAddress
Compiling class xobt.soap.TesterWebServiceHttpPort.getSystemProperties
Compiling routine xobt.soap.TesterWebServiceHttpPort.doArrayInput.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doBoolean.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doCalendar.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doDate.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doDouble.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doEcho.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doEchoEmployeeInfoVO.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doEchoXml.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doFloat.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doInt.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doLong.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doPing.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doShort.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.doSleep.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.getDOI.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.getEmployeeInfo.1
```

```
Compiling routine xobt.soap.TesterWebServiceHttpPort.getEmployeeInfoVO.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.getEmployeeList.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.getEmployeeListOnly.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.getGettysburgAddress.1
Compiling routine xobt.soap.TesterWebServiceHttpPort.getSystemProperties.1
Compilation finished successfully.

 o  WEB SERVICE 'XOBT TESTER WEB SERVICE' addition/update succeeded.

 o  Web service client classes created successfully.
 o  WEB SERVICE 'XOBT TESTER REST SERVICE' addition/update succeeded.

 o  added/updated XOBT SAMPLE SERVER lookup key.

 Updating Routine file...

 Updating KIDS files...

 XOBT 1.0 Installed.
               Jun 27, 2007@11:11:29

 Install Message sent

Install Completed
```

Appendix D: Testing ^%SYS("TempDir") Setting

Caché allows the designation of a temporary directory via the ^%SYS("TempDir") global node. Various Caché classes will use the value of this global node, if set, as the directory location to store temporary files in, such as stream files used by its web service client classes.

The ^%SYS global is unique to each Caché instance/node, so the ^%SYS("TempDir") setting must be set individually on each instance/node.

To **test** whether ^%SYS("TempDir") is working correctly for Caché classes on a particular box/instance:

1. Log into any one of your Caché boxes/instances, using a typical end-user account.

2. Go into programmer mode.

3. Enter the following M commands at the programmer prompt to try to create a temporary file:

   ```
   K %objlasterror
   S x=##class(%Library.FileBinaryStream).%New()
   DO x.WriteLine("test test test")
   DO $system.OBJ.DisplayError(%objlasterror)
   ```

   If the variable *%objlasterror* is reported as undefined by $systeml.OBJ.DisplayError(), then continue. Otherwise, if an error is displayed, e.g.:

   > ERROR #5005: Cannot open file ' WSERPRQNG6H.stream'

   In that case, stop here and examine the settings for ^%SYS("TempDir") on this system, and adjust them if ^%SYS("TempDir") is either undefined, pointing to a non-existent directory, or pointing to a directory without W:RWED privileges. Then try this test again from the beginning.

4. If you didn't get an error in the previous step, continue and enter the following command:

   ```
   DEVMOU>W x.Filename<Enter>
     USER$:[TEMP]WSERPRQNG6H.stream
   ```

   Confirm that the directory listed as part of the stream filename is the expected directory that ^%SYS("TempDir") is set to. If no directory is present, examine the setting of ^%SYS("TempDir") on this system.

5. Continue and enter the following command:

```
DEVMOU>DO x.OutputToDevice()<Enter>

test test test

DEVMOU>
```

Verify that output ('test test test') is written to the screen. This confirms that the stream file was able to hold data written to it earlier, that it can now retrieve and write back.

6. Continue and enter the following commands:

```
S sc=x.Clear()
DO $system.OBJ.DisplayError(%objlasterror)
```

This proves whether delete privileges are present to the temporary directory. If the variable *%objlasterror* is reported as undefined by $systeml.OBJ.DisplayError(), the test on this system is complete. The temporary directory protection settings are sufficient, on this box/instance, for the particular end-user account used.

Otherwise, if an error is displayed, e.g.:

ERROR #5019: Cannot delete file ' WSERPRQNG6H.stream'

In that case, stop here and examine the settings for ^%SYS("TempDir") on this system, and adjust them if ^%SYS("TempDir") is either undefined, pointing to a non-existent directory, or pointing to a directory without W:RWED privileges. Then try this test again from the beginning.