



Xcentrity® BIS AddPack for Visual COBOL 2.2 for UNIX

A large, decorative graphic consisting of multiple overlapping, wavy blue lines that create a sense of motion and depth. The lines are in various shades of blue, from dark to light, and are set against a light blue gradient background.

Release Notes

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

Copyright © Micro Focus 2009-2013. All rights reserved.

MICRO FOCUS, the Micro Focus logo and Visual COBOL are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.

All other marks are the property of their respective owners.

2013-10-08

Contents

Xcentricity® BIS AddPack for Visual COBOL 2.2 for UNIX	4
Installation	5
Installing the UNIX Product	5
System Requirements	5
Packaging	5
Installation	5
Verifying the Installation	7
After Installing	11
Environment Variables	11
Xcentricity BIS Tutorial	11
Configuring Applications (UNIX only)	12
Uninstalling	13
Uninstalling the Request Handler	13
Uninstalling the Service Engine	13

Xcentrisity[®] BIS AddPack for Visual COBOL 2.2 for UNIX

These release notes contain information that might not appear in the Help. Read them in their entirety before you install the product.

Installation

Installing the UNIX Product

System Requirements

Operating Systems Supported

This AddPack supports the following operating systems:

- AIX 6.1
- AIX 7.1
- Red Hat Linux
- SUSE Linux

Additional System Prerequisites

Apache 2.2 HTTP Server must be installed. Note that the Apache server may be either a 32-bit program or a 64-bit program. It must allow modules which are Dynamic Shared Objects to be loaded.

This AddPack requires one of the following Micro Focus products:

Visual COBOL Development Hub (DevHub), version 2.1 update 1
COBOL Server version 2.1 update 1

Earlier versions of Visual COBOL and COBOL Server are not supported and will not work.

Packaging

There is one edition of Xcentrisity BIS for Visual COBOL. It installs both 32-bit and 64-bit versions. However, when configuring Xcentrisity BIS, the selected bit-size of Xcentrisity BIS must match the bit-size of the Apache HTTP Server.

Installation

To install Xcentrisity BIS for Visual COBOL on UNIX::

1. Download this AddPack from the Product Updates pages on Micro Focus SupportLine and save the file to your computer.
2. Make the file executable, using the `chmod` command.
3. Run the file, with the `-installlocation` option set to the location where Visual COBOL or COBOL Server is installed.

Before using Xcentrisity BIS for Visual COBOL, you must configure it. There are two parts to BIS, and each must be configured separately.

Configuring the Service Engine

The Service Engine is the part of Xcentrisity BIS for Visual COBOL that runs COBOL programs. It is a pair of UNIX daemons that are started automatically when the operating system is started into multi-user mode. The Service Engine starts a COBOL program when it receives a request from the Request Handler. The

COBOL program runs until it chooses to terminate or until the Request Handler informs the Service Engine that the program is no longer needed.

Configuring the Service Engine creates the settings file that the Service Engine reads when starting. It also places files in the `/etc` directory that allow the Service Engine to start automatically.

To start the Service Engine configuration:

1. Open a terminal window. Be sure that you are running as the root user.
2. Set `COBDIR` to the location of the installed Visual COBOL or COBOL Server product:

```
. install-location/bin/cobsetenv
```

3. Start the script to perform the configuration:

```
`${COBDIR}/bin/config_bis_daemons options
```

You can use these options in the `config_bis_daemons` script:

- `-b bits`, where `bits` is either 32 or 64. This selects the 32-bit or 64-bit Service Engine. If the machine supports both 32-bit and 64-bit programs, the default is 64. Otherwise, the default is the supported size.
- `-e` requests that the configuration script prompt the user for the values of the settings that control the behavior of the Service Engine.
- `-h` prints a help message.
- `-q` requests that the configuration script display fewer and briefer messages.
- `-w` specifies a location where the BIS Service Engine and Request Handler creates files while running. The default value is `/var/local/xbis`. The configuration script creates this directory if it does not exist.

Settings are stored in a file created in the ``${COBDIR}/etc` directory. This file is copied to `/etc/xbis.conf`, where the BIS Service Engine looks for it. This file may be edited with a text editor to change any of the settings. The Service Engine must be stopped and restarted if running for the new settings to take effect.



Note: The most overlooked setting in the `/etc/xbis.conf` file is the `UserName` setting. Although the Service Engine daemons run as the root user, the processes used by the COBOL programs use the value of this setting. This user ID must exist on the computer at the time the Service Engine is started, or the Service Engine will not start. The default value of this setting is `bis`. You should either set the value as `bis` or change this setting to the appropriate user ID.

When the configuration script completes, it gives the command to start the Service Engine immediately. The Service Engine should also start automatically whenever the operating system starts.



Tip: You should record this command, as it can be used to manually start, stop, or restart the Service Engine.

The file `/etc/sysconfig/xbis` contains shell commands to set environment variables. It is read when the Service Engine is started. However, this is rarely needed.

Configuring the Request Handler

The Request Handler is the part of Xcentrity BIS for Visual COBOL that processes BIS requests sent to the Apache HTTP Server. It is an Apache module in the form of a Shared Object Library. It processes requests directed to BIS stencil files. A stencil files have an `.srf` filename extension.

Configuring the Request Handler creates a file named `mod_xbis.conf` in the ``${COBDIR}/etc` directory. This file contains Apache configuration commands. After the configuration script runs, add this file to the Apache HTTP Server configuration.

You can use these options in the `config_bis_apache` script:

- `-b bits`, where *bits* is either 32 or 64. This selects the 32-bit or 64-bit Service Engine. If the machine supports both 32-bit and 64-bit programs, the default is 64. Otherwise, the default is the supported size.
- `-h` prints a help message.
- `-w` specifies a location where the BIS Service Engine and Request Handler creates files while running. The default value is `/var/local/xbis`.

The `-b` and `-w` options on the `config_bis_apache` script should be the same as the `-b` and `-w` options on the `config_bis_daemons` script.

The options and the value of `$COBDIR` are used to create a customized file named `$COBDIR/etc/mod_xbis.conf`. This is the file you customize for your application. It is an Apache configuration file. For documentation of Apache configuration files, see <http://httpd.apache.org/docs/2.2/configuring.html>.

When the script completes, it displays instructions on adding the `mod_xbis.conf` file to your Apache configuration. In most cases, this instructs you to edit the main Apache configuration file and add an `Insert` directive that names the `mod_xbis.conf` file. If Apache was built from source, the Apache configuration file is named `httpd.conf` and is in the Apache `conf` directory.

You may also need to add the `$COBDIR/lib` directory to the library path environment variable in an Apache file named `envvars`, typically in the Apache `bin` directory. Read all of the instructions given by the script.

After editing the Apache files needed to add the BIS Request Handler to Apache, start or restart the Apache HTTP Server. This may be done using the `apachectl` command.

The created `mod_xbis.conf` file contains comments to help explain the purpose of the directives in the file. As initially created, it contains directives to associate the URL path `/xbisvc21/samples` with the directory `$COBDIR/demo/BIS/Xcentrisity`.

You may use the directives that define the location of the BIS samples as examples. There is a commented section near end of the `mod_xbis.conf` file that is entitled `/app1`. It is intended to serve as a model for your application. Note the following:

- You need not start the URL path with `/xbisvc21` or `/app1`.
- You can set environment variables for a particular directory, as illustrated by the examples.
- Environment variables set in the `mod_xbis.conf` file, whether set globally or for a single directory, are available for use in stencil files.
- Environment variables set in a stencil file are available to the COBOL program.

Verifying the Installation

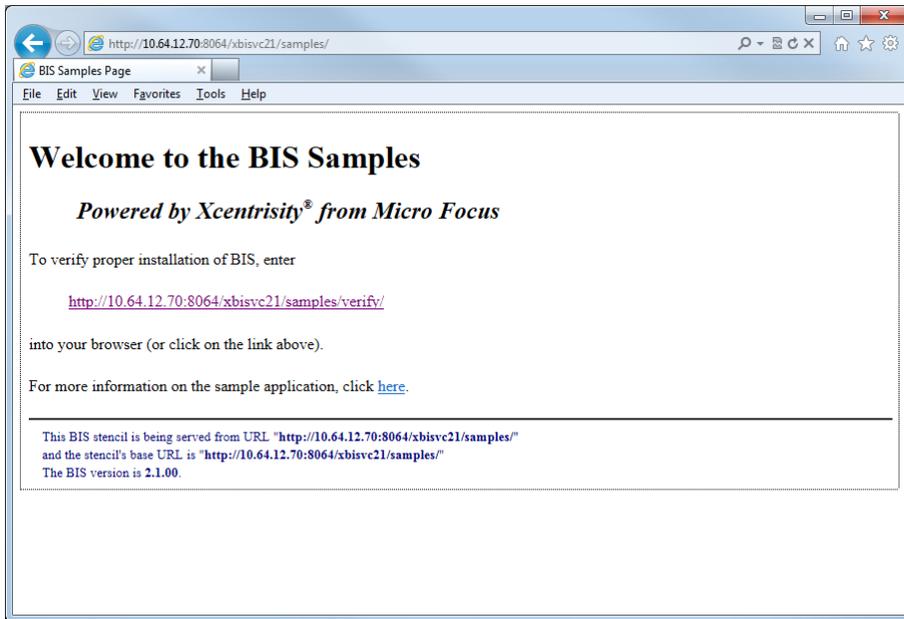
To verify that Xcentrisity BIS for Visual COBOL was correctly installed, run the `VERIFY` sample program.

In a browser window, type the URL where the samples reside:

```
http://server:port/xbisvc21/samples/
```

In this URL, `server` is the name of the server on which Xcentrisity BIS for Visual COBOL is installed, and `port` is port number on which the Apache HTTP Server is listening. If the Apache server is listening on port 80 (the default), then the colon and the port number may be omitted.

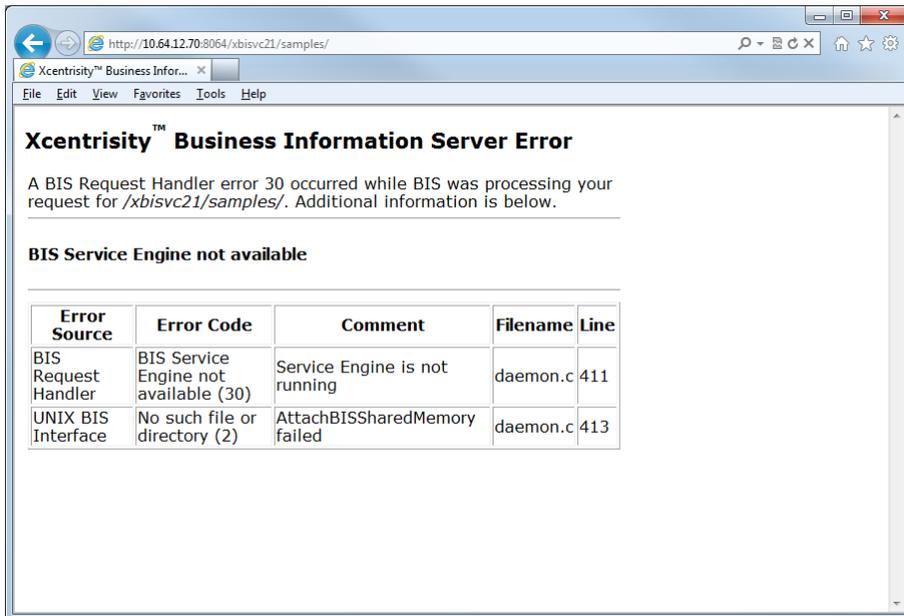
The request is sent to the BIS server, which returns the **Welcome to the BIS Samples** page.



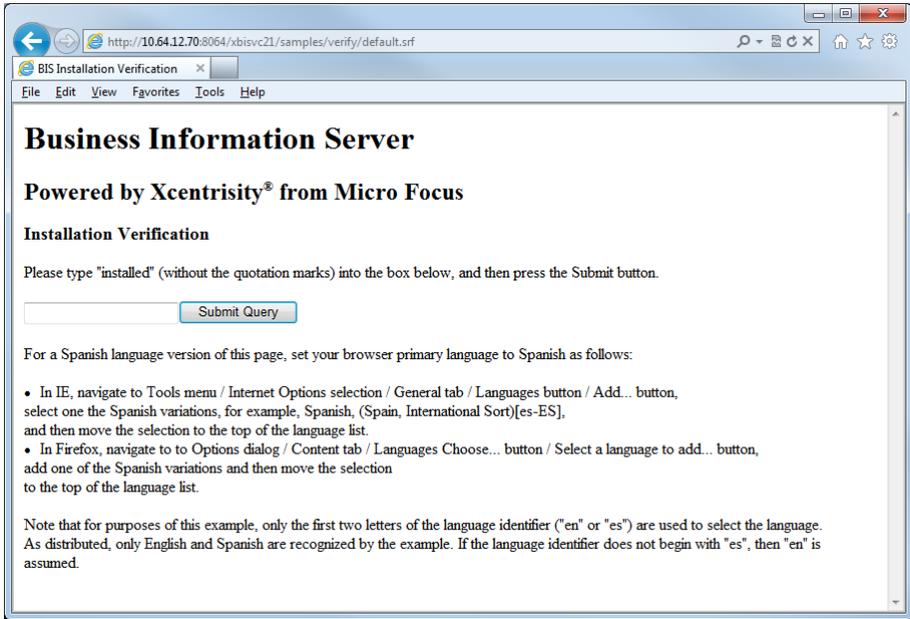
If the **Welcome to the BIS Samples** page does not appear, then one of the following conditions exist:

- Xcentrity BIS, or a component that BIS depends on is not installed correctly.
- The BIS Service Engine was not configured properly.
- The Apache HTTP Server was not configured to use the BIS Request Handler.

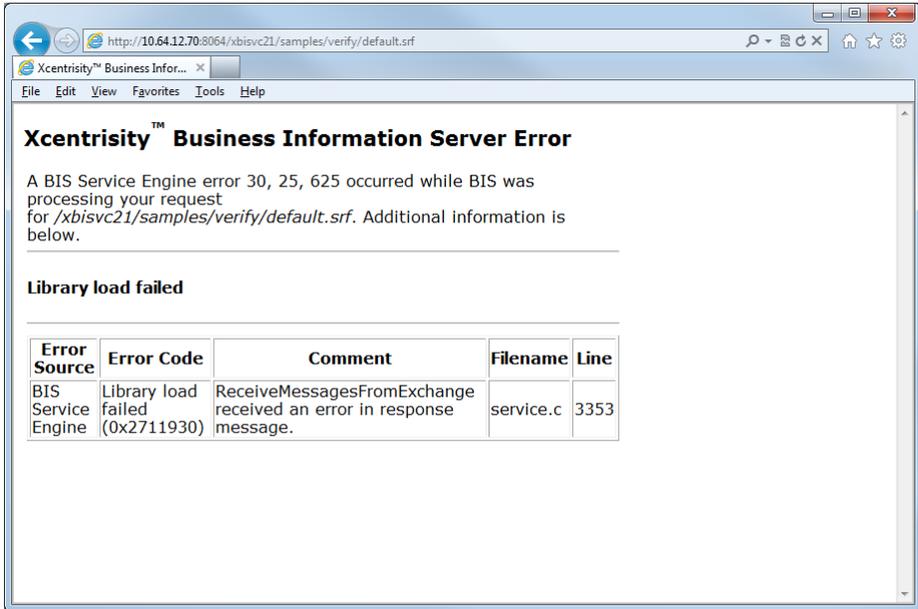
This page indicates that the Xcentrity BIS Service Engine is not running:



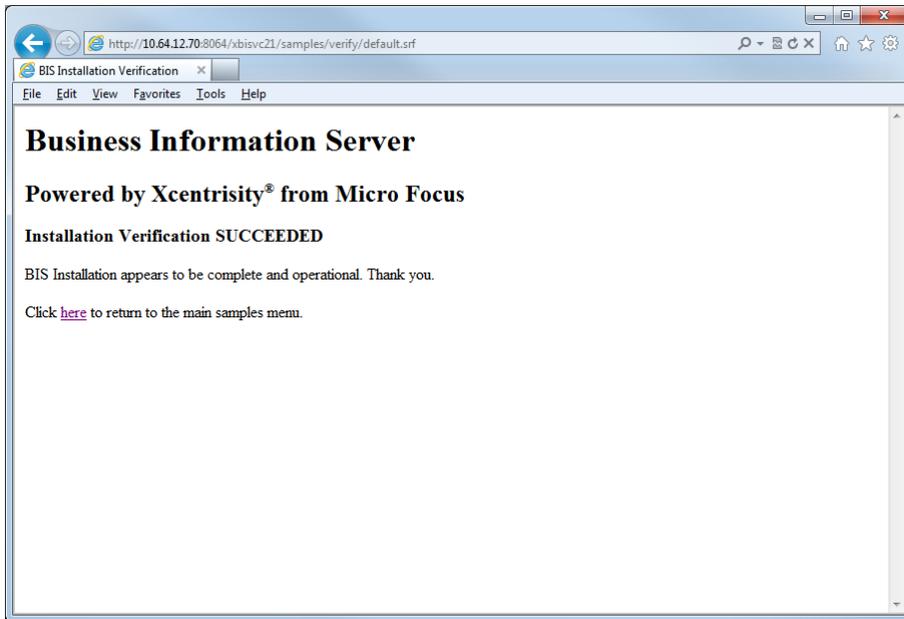
If the **Welcome to the BIS Samples** page appears, click the URL link in the Error page that ends in "verify". The **BIS Installation Verification** page should appear.



If an error page appears, it indicates that the Visual COBOL service program cannot be started. For example, the error code, `Library load failed` in the following page indicates that an object library file cannot be loaded. In this case, the file containing the COBOL program could not be loaded because it was a 32-bit COBOL program and the server was running the 64-bit version of BIS.



If the **Installation Verification** page appears, type `Installed` in the text box and click **Submit Query**. The following page should appear to confirm correct operation:



To turn on visible tracing of the verify sample, change the URL to:

`http://server:port/xbisvc21/samples/verify/default.srf?trace=page`

To diagnose such a problem, examine the BIS trace output (if available), and be sure that Visual COBOL programs can be run outside the BIS environment.

You can also use the Visual COBOL Consolidated Tracing Facility (CTF) to diagnose problems. You can set the `MFTRACE_CONFIG` and `MFTRACE_LOGS` environment variables in the stencil file (`.srf` file) of the page with the problem. Be sure that the `MFTRACE_LOGS` environment variable points at a directory which is writable by the user selected in `/etc/xbis.conf`.

After Installing

Reference information for Xcentricity BIS is in a PDF file that is installed with the file name `$COBDIR/docs/BISUsersGuideVC.pdf`.

All COBOL-callable functions for BIS have been renamed for this version of Xcentricity BIS. In the *User's Guide*, the names of these functions all begin with the characters B\$. These have all been changed to B_. Thus, they are `B_ReadRequest`, `B_WriteResponse`, `B_SetInactivityTimeout`, `B_SetServiceTimeout`, `B_SetResponseStatus`, and `B_Trace`.

Environment Variables

Certain environment variables are defined for use in BIS stencil files. These are set by Apache directives in `$COBDIR/etc/mod_xbis.conf`. The following are defined at the beginning of the file, and are available to all BIS applications:

- `SIXFOUR` is defined to be the two-character string 64 if the 64-bit version of BIS was selected when the Request Handler was configured. Otherwise, it is defined as an empty string.
- `COBDIR` is defined to be the location where Visual COBOL and Xcentricity BIS are installed. It is the value of the `COBDIR` environment variable at the when the `config_bis_apache` script was run.

You may add definitions of other environment variables to the `mod_xbis.conf` file, either globally or for a single directory. They will then also be available in BIS stencil files.

To use these variables in a stencil file, close the environment variable name in `%` characters. See the stencil files in the `verify` sample and the `tutorial1` sample.

To make environment variables available to COBOL programs, use the `{{SetEnv}}` tag in the BIS stencil file. For example, the `default.srf` stencil file in the `verify` sample contains the following tag:

```
{{ SetEnv(COBPATH=.:%COBDIR%/bin:%COBDIR%/lib) }}
```

This sets the `COBPATH` variable in the environment of the COBOL program. Note that it uses the `COBDIR` variable that was set in `$COBDIR/etc/mod_xbis.conf`.

Xcentricity BIS Tutorial

The best way to learn about Xcentricity BIS is to follow the BIS Tutorial. A PDF file that documents the BIS Tutorial is installed with the file name `$COBDIR/docs/BIS_Tutorial.pdf`. You should read this document..

Before running the BIS Tutorial, you must ensure that the directory where the BIS Tutorial is located is writable by the user ID that BIS uses to run COBOL programs. This user ID was specified by the Service Engine configuration. It may be found in the file `/etc/xbis.conf`.

The BIS Tutorial is installed at `$COBDIR/demo/BIS/Xcentricity/tutorial1`. You must ensure that this directory and the data subdirectory are writable by the BIS user ID. The easiest way to do this to make both directories writable by all users. The following is one way to do this:

```
chmod -R a+w $COBDIR/demo/BIS/Xcentricity/tutorial1
```

Configuring Applications (UNIX only)

To configure an application to run under Xcentrinity BIS, you need to place a stencil file (with an `.srf` file extension) into a directory which the Apache HTTP Server can access. You must add Apache directives to the Apache configuration that give the name of this directory and the URL by which clients can reach it.

Most commonly, the Apache directives are placed in `$COBDIR/etc/mod_xbis.conf`. The `/appl` section of the configured `mod_xbis.conf` file can serve as a model. It is also possible to place these directives directly in the main Apache configuration file. They may even be placed in a user-created file, as long as the Apache configuration includes an Include directive that names the file.

See the [Apache HTTP Server Configuration Files document](#) for detailed information about Apache configuration files. For most users, all that is needed to define an application directory in Apache is given in the `/appl` section of the configured `mod_xbis.conf` file. These directives are:

- **Alias** associates a URL path with directory in the file system.
- **<Directory path> ... </Directory>** encloses directives that apply only the named directory. This is needed if the directory is not part of the Apache “DocumentRoot”. The following directives should be within a **<Directory> ... </Directory>** pair.
- **DirectoryIndex** can be used to specify the name of the default file to be used when a client requests a directory. Although it can be used for other purposes, BIS applications usually use it to name the default BIS stencil file in the directory.
- **Order** controls the order in which Apache processes Allow and Deny directives.
- **Allow** describes the clients which are allowed to access an area of the server. The samples use “Allow from all” to allow unrestricted access.
- **Deny** describes the clients which are prohibited from accessing an area of the server.

In the installed BIS samples, all of the files for the sample are placed under a single directory. This is not a requirement. The only file that needs to be in the application directory is the stencil file (`.srf` file). Note that when the COBOL program runs, the current directory becomes the directory containing the stencil file. However, by setting the `COBPATH` and `COBDATA` environment variables in the stencil file, the program and data files may be located elsewhere.

Uninstalling

You must separately uninstall the Xcentrity BIS Request Handler and Service Engine. Uninstall the Request Handler first. You must be running as `root` to uninstall Xcentrity BIS.

Uninstalling the Request Handler

To uninstall the Request Handler, edit the Apache files that refer to BIS. These are the files that are edited when configuring the Request Handler. These are:

- The Apache configuration file, which should contain an `Insert` directive that refers to the `mod_xbis.conf` file located in the `$COBDIR/etc` directory. If Apache was built from source, this file is named `httpd.conf` and is in the Apache `conf` directory.
- To add the `$COBDIR/lib` directory to the library path environment variable, undo your changes to the Apache `envvars` file (in the Apache `bin` directory).

Restart Apache after removing references to Xcentrity BIS from the Apache HTTP Server using Apache's `apachectl` command.

Uninstalling the Service Engine

To uninstall the Xcentrity BIS Service Engine, you must first stop the Service Engine. You must be running as `root` to do this.

The normal way to stop the Service Engine is to use the command that was given at the end of the configuration of the Service Engine. It will be a script located in a subdirectory of `/etc`. The exact location is dependent of the operating system.

An alternative way to stop the Service Engine is to use the `$COBDIR/bin/xbisctl` or `$COBDIR/bin/xbisctl64` command, depending on whether the 32-bit or 64-bit version of the Service Engine is running. To do this, enter the command as follows:

```
$COBDIR/bin/xbisctl stop
```

After stopping the Service Engine, remove the files that start the Service Engine whenever the operating system restarts. To simplify this, the Service Engine configuration script creates a small shell script named `$COBDIR/bin/uninstall_bis_daemons`. Enter the following to run this script:

```
$COBDIR/bin/uninstall_bis_daemons
```

This removes the script that starts the Service Engine when the operating system starts and stops the Service Engine when the operating system shuts down. It also removes the `/etc/xbis.conf` file. It removes the `bislogs` directory from the BIS working directory. Lastly, it removes the BIS working directory, if it is empty. If the BIS working directory is not empty, remove it manually after verifying that you do not want any of its contents