STAF Installation Guide

December 16, 2013

This document describes how to install STAF V3.4.16 and later.

To find more detailed information on using STAF, go to the main STAF web page

1. Introduction

1.1. Installing STAF

There are two types of installers that are provided for STAF:

- InstallAnywhere (available for Windows and most Unix platforms)
- STAFInst (available for all Unix platforms)

Note that the InstallAnywhere and STAFInst installers install the exact same files on a given platform. You can select which type of installer is most appropriate to use in your environment.

1.2. InstallAnywhere

InstallAnywhere is a Java-based multi-platform software installation program.

InstallAnywhere provides three installation modes:

- Graphical installation mode (requires a UI display)
- Console installation mode (via command-line, useful for systems without a UI display but requiring an interactive install)
- Silent installation mode (where the install options are specified when starting the silent installation, either via command-line options or a response file)

Note that if you start a graphical installation, but the system does not hava a display (i.e. if you are accessing a Unix system via telnet or ssh), the installer will detect this situation and continue the installation in console mode.

There are three types of InstallAnywhere files that are provided for STAF:

- Installer executable with a bundled JVM (named **STAF<version>-setup-<platform>.exe** on Windows and **STAF<version>-setup-<platform>.bin** on Unix)

  The bundled JVM will be used for the installation, and will be copied to the "jre" directory in the root STAF install directory. This JVM will also be used to uninstall STAF.

- Installer executable without a bundled JVM (named **STAF<version>-setup-<platform>-NoJVM.exe** on Windows, **STAF<version>-setup-<platform>-NoJVM.bin** on Unix, and **STAF<version>-setup-<platform>.bin** on Mac OS X)

- Installer zip file without a bundled JVM for Mac OS X (named **STAF<version>-setup-<platform>.zip**)

To run an installer without a bundled JVM, you must already have a JVM on the system and accessible via the PATH. Supported JVMs are:

- Sun: 1.5.x, 1.6.x, 1.7.x
- IBM: 1.5.x, 1.6.x, 1.7.x
- Apple: 1.5.x, 1.6.x
- HP: 1.5.x, 1.6.x
- Oracle: 1.7.x

If you get the following error when running an installer without a bundled JVM, then that means that InstallAnywhere could not find a valid JVM:

```
C:\temp>STAF3416-setup-win32-NoJVM.exe
LaunchAnywhere Error: Could not find a valid Java virtual machine to load.
You may need to reinstall a supported Java virtual machine.
```

You can add LAX_VM "full-path-to-java-exe" as parameters to the STAF NoJVM installer to force the installer to run against a specific Java executable that you have installed on your machine. For example:

```
STAF3416-setup-win32-NoJVM.exe LAX_VM "C:\Program Files\Java\jre6\bin\java.exe"
```

To use an InstallAnywhere STAF installer, the target system must have:

- 64 MB of free RAM
- Minimum of 8-bit color depth (256 colors) for graphical installation
- Minimum 640 X 480 screen resolution for graphical installation

On Unix systems, after downloading the InstallAnywhere STAF installer, you must change the file permissions to make it executable before running it. For example:

```
chmod +x STAF3416-setup-linux.bin
./STAF3416-setup-linux.bin
```

An InstallAnywhere install of STAF will typically require 1-5 minutes depending on the platform and system performance.

More information on InstallAnywhere can be found at http://www.flexerasoftware.com/products/installanywhere.htm

1.3. STAFInst

STAFInst is a script-based installer for Unix platforms. It will install all files required to run STAF, however, it will not perform some installation steps that InstallAnywhere performs, such as updating user profiles for environment variable updates.

The STAFInst installer is packaged as a "GNU zipped tar" file (named **STAF<version>-<platform>.tar.gz**) except on z/OS where it is packaged as a Unix compressed file (named **STAF<version>-<platform>.tar.Z**).

A STAFInst install of STAF will typically require about 1 minute.

2. Installers available for each platform

2.1. Overview

To download the latest versions of STAF, go to the Download STAF page.

The following table shows an overview of the STAF installers available for each platform.

Platform	InstallAnywhere (Bundled JVM)	InstallAnywhere (No JVM)	STAFInst
Windows 32-bit (win32)	✔	✔	-
Windows AMD64/Opteron (winamd64)	✔	✔	-
Linux Intel 32-bit (linux)	✔	✔	✔
Linux AMD64/Opteron/x86_64 (linux-amd64)	✔	✔	✔
Linux PPC64-32 (linux-ppc64-32)	✔	✔	✔
Linux PPC64-64 (linux-ppc64-64)	✔	✔	✔
zLinux 31-bit (zlinux-32)	✔	✔	✔
zLinux 64-bit (zlinux-64)	✔	✔	✔
z/OS 32-bit (zos)	-	-	✔
z/OS 64-bit (zos64)	-	-	✔
Solaris Sparc 32-bit (solaris-sparc)	✔	✔	✔
Solaris Sparc 64-bit (solaris-sparc64)	✔	✔	✔

Solaris AMD64/Opteron x64 32-bit Java (solaris-x64)	✔	✔	✔
Solaris AMD64/Opteron x64 64-bit Java (solaris-x64-64)	✔	✔	✔
Solaris x86 32-bit (solaris-x86)	✔	✔	✔
AIX 32-bit (aix)	✔	✔	✔
AIX 64-bit (aix64)	✔	✔	✔
HPUX PA-RISC 32-bit (hpux)	✔	✔	✔
HPUX PA-RISC 64-bit (hpux-parisc64)	✔	✔	✔
HPUX IA64-32 (hpux-ia64-32)	✔	✔	✔
HPUX IA64-64 (hpux-ia64-64)	✔	✔	✔
IBM i 32-bit, previously known as i5/OS or OS/400 (aix)	-	-	✔
IBM i 64-bit, previously known as i5/OS or OS/400 (aix64)	-	-	✔
FreeBSD i386 (freebsd)	-	✔	✔
Mac OS X 10.4+ i386 (macosx-i386)	-	✔	✔
Mac OS X 10.4+ PPC (macosx-ppc)	-	✔	✔
Mac OS X 10.6+ Universal binary for i386, x86_64, and ppc (macosx-universal)	-	✔	✔

2.2. Install properties for STAF Installers

After the STAF install is complete, an install.properties file will be created in the root STAF install directory. The file will contain key/value pairs that provide information about the version of STAF that has been installed.

The install.properties file will contain the following information:

- **version** - the version of STAF that has been installed
- **platform** - the STAF platform name
- **architecture** - the architecture of the STAF build (32-bit or 64-bit)
- **installer** - the type of installer (InstallAnywhere, STAFInst)
- **file** - the file used to install STAF
- **osname** - the operating system name for the STAF build (equivalent to the "os.name" Java property)
- **osversion** - the operating system version supported by the STAF build ("*" indicates the build is supported on any version of the OS; a version number followed by a "+" indicates the build supports that version or later)
- **osarch** - the operating system architecture supported by the STAF build (equivalent to the "os.arch" Java property)

The following table shows the possible values for the install.properties file

version	platform	architecture	installer	file	osname	osversion	osarch
3.x.x	win32	32-bit	IA | IA_NoJVM	STAF3xx-setup-win32.exe | STAF3xx-setup-win32-NoJVM.exe	Windows	*	x86
3.x.x	winamd64	64-bit	IA | IA_NoJVM	STAF3xx-setup-winamd64.exe | STAF3xx-setup-winamd64-NoJVM.exe	Windows	*	amd64
3.x.x	linux	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-linux.bin | STAF3xx-setup-linux-NoJVM.bin | STAF3xx-linux.tar.gz	Linux	*	x86
3.x.x	linux-amd64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-linux-amd64.bin | STAF3xx-setup-linux-amd64-NoJVM.bin | STAF3xx-linux-amd64.tar.gz	Linux	*	amd64
3.x.x	linux-ppc64-32	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-linux-ppc64-32.bin | STAF3xx-setup-linux-ppc64-32-NoJVM.bin | STAF3xx-linux-ppc64-32.tar.gz	Linux	*	ppc64
3.x.x	linux-ppc64-64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-linux-ppc64-64.bin | STAF3xx-setup-linux-ppc64-64-NoJVM.bin | STAF3xx-linux-ppc64-64.tar.gz	Linux	*	ppc64

3.x.x	zlinux-32	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-zlinux-32.bin | STAF3xx-setup-zlinux-32-NoJVM.bin | STAF3xx-zlinux-32.tar.gz	Linux	SLES10+/RHEL5+	s390
3.x.x	zlinux-64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-zlinux-64.bin | STAF3xx-setup-zlinux-64-NoJVM.bin | STAF3xx-zlinux-64.tar.gz	Linux	SLES10+/RHEL5+	s390x
3.x.x	zos	32-bit	STAFInst	STAF3xx-zos.tar.Z	z/OS	1.4+	390
3.x.x	zos64	64-bit	STAFInst	STAF3xx-zos64.tar.Z	z/OS	1.4+	s390x
3.x.x	solaris-sparc	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-solaris-sparc.bin | STAF3xx-setup-solaris-sparc-NoJVM.bin | STAF3xx-solaris-sparc.tar.gz	SunOS	5.10+	sparc
3.x.x	solaris-sparc64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-solaris-sparc64.bin | STAF3xx-setup-solaris-sparc64-NoJVM.bin | STAF3xx-solaris-sparc64.tar.gz	SunOS	5.10+	sparcv9
3.x.x	solaris-x64	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-solaris-x64.bin | STAF3xx-setup-solaris-x64-NoJVM.bin | STAF3xx-solaris-x64.tar.gz	SunOS	5.10+	x86-64 (aka x64, amd64)

3.x.x	solaris-x64-64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-solaris-x64-64.bin | STAF3xx-setup-solaris-x64-64-NoJVM.bin | STAF3xx-solaris-x64-64.tar.gz	SunOS	5.10+	amd64
3.x.x	solaris-x86	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-solaris-x86.bin | STAF3xx-setup-solaris-x86-NoJVM.bin | STAF3xx-solaris-x86.tar.gz	SunOS	5.10+	x86-32 (aka x86)
3.x.x	aix	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-aix.bin | STAF3xx-setup-aix-NoJVM.bin | STAF3xx-aix.tar.gz	AIX	6.1+	ppc
3.x.x	aix64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-aix64.bin | STAF3xx-setup-aix64-NoJVM.bin | STAF3xx-aix64.tar.gz	AIX	6.1+	ppc64
3.x.x	hpux	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-hpux.bin | STAF3xx-setup-hpux-NoJVM.bin | STAF3xx-hpux.tar.gz	HP-UX	B.11.11+	PA_RISC2.0
3.x.x	hpux-parisc64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-hpux-parisc64.bin | STAF3xx-setup-hpux-parisc64-NoJVM.bin | STAF3xx-hpux-parisc64.tar.gz	HP-UX	B.11.11+	PA_RISC2.0W

3.x.x	hpux-ia64-32	32-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-hpux-ia64-32.bin | STAF3xx-setup-hpux-ia64-32-NoJVM.bin | STAF3xx-hpux-ia64-32.tar.gz	HP-UX	B.11.23+	IA64N
3.x.x	hpux-ia64-64	64-bit	IA | IA_NoJVM | STAFInst	STAF3xx-setup-hpux-ia64-64.bin | STAF3xx-setup-hpux-ia64-64-NoJVM.bin | STAF3xx-hpux-ia64-64.tar.gz	HP-UX	B.11.23+	IA64W
3.x.x	IBM i 32-bit	32-bit	STAFInst	STAF3xx-aix.tar.gz	AIX	AIX 6.1+, IBM i 7.1+	ppc
3.x.x	IBM i 64-bit	64-bit	STAFInst	STAF3xx-aix64.tar.gz	AIX	AIX 6.1+, IBM i 7.1+	ppc64
3.x.x	freebsd	32-bit	IA_NoJVM | STAFInst	STAF3xx-setup-freebsd-NoJVM.bin | STAF3xx-freebsd.tar.gz	FreeBSD	7.4+	i386
3.x.x	macosx-i386	32-bit	IA_NoJVM | STAFInst	STAF3xx-setup-macosx-i386.zip | STAF3xx-setup-macosx-i386.bin | STAF3xx-macosx-i386.tar.gz	Mac OS X	10.4+	i386
3.x.x	macosx-ppc	32-bit	IA_NoJVM | STAFInst	STAF3xx-setup-macosx-ppc.zip | STAF3xx-setup-macosx-ppc.bin | STAF3xx-macosx-ppc.tar.gz	Mac OS X	10.4+	ppc

3.x.x	macosx-universal	32-bit/64-bit	IA_NoJVM | STAFInst	STAF3xx-setup-macosx-universal.zip | STAF3xx-setup-macosx-universal.bin | STAF3xx-macosx-universal.tar.gz	Mac OS X	10.6+	universal (i386, x86_64, ppc)

Note that the install.properties file will be overwritten every time you install STAF to a target location. You should not delete or modify the install.properties file.

Here is a sample install.properties file from a Windows system (using the IA installer):

```
version=3.4.16
platform=win32
architecture=32-bit
installer=IA
file=STAF3416-setup-win32.exe
osname=Windows
osversion=*
osarch=x86
```

Here is a sample install.properties file from a Mac OS X i386 system (using the STAFInst installer):

```
version=3.4.16
platform=macosx-universal
architecture=32-bit/64-bit
installer=STAFInst
file=STAF3416-macosx-universal.tar
osname=Mac OS X
osversion=10.6+
osarch=universal (i386, x86_64, ppc)
```

Note that if STAF is running on a system, you can use the "MISC LIST PROPERTIES" service request to obtain the information in the install.properties file

When using Java classes that call into the STAF APIs, it is important to understand that STAF is implemented in C++, so any calls that your Java code makes into the STAF APIs will be using JNI to interface with the native C++ code.

This means that you must use the version of STAF that is compatible with the architecture (32-bit or 64-bit) of the JVM that you are using.

If the "STAF architecture" is "32-bit", then you must use a 32-bit JVM with that version of STAF.

If the "STAF architecture" is "64-bit", then you must use a 64-bit JVM with that version of STAF.

3. Install Topics

3.1. STAF License Agreement

In order to install STAF, you must read and accept the terms of the license agreement.

3.2. Installation Target

When installing STAF, you must specify the target installation directory. The default install location varies depending on the operating system.

- On Windows, the default install directory is **C:\STAF**
- On Mac OS X, the default install directory is **/Library/staf**
- On other Unix platforms, the default install directory is **/usr/local/staf**

3.3. Upgrading STAF

When installing STAF via InstallAnywhere, if you have selected a target installation directory where a version of STAF is already installed, you will be asked if you want to upgrade the existing version. If you choose to do the upgrade installation, all of STAF files will be upgraded to the new version. Any files created after STAF was installed (e.g. additional services, log files, updated STAF.cfg, etc) will not be removed. Note that when upgrading STAF, the same installation options are selected by default as when doing a new install. That is, a STAF upgrade does not use the settings that were selected by the previous STAF install.

When installing STAF via STAFInst, if you have selected a target installation directory where a version of STAF is already installed, the install will display an error message indicating that you must first uninstall the existing version of STAF.

3.4. Install Sets

There are several install sets available when installing STAF, which define the set of STAF features that will be installed:

- Full - All features will be installed. This option is recommended for most users.
- Minimal - Only required application features will be installed. This option is recommended only for users with limited disk space.
- Custom - Choose this option to customize the features to be installed. This install set is only available with the InstallAnywhere installers.

Here is a list of the available features:

- STAF Application (STAF) - Core STAF application
- External Services (ExtSvcs) - External STAF services (Log, Monitor, ResPool, Zip)
- Language Support (Langs) - STAF Language Support (Java, Perl, Python, TCL, etc.)
- Samples and Demos (Samples) - Samples and Demos
- Additional Codepage Support (Codepage) - Install all STAF codepages
- Documentation (Docs) - STAF documenation
- Development Support (Develop) - Service developer header files and libraries

Note that the STAFInst installers allow install set types of "recommended", "full", and "minimal". Selecting "recommended" or "full" will perform a full install.

3.5. Registration

During the install you can optionally choose to provide registration information to the STAF development team. This information is used by the STAF team to determine the number of projects that are using STAF.

Note that this registration information is currently transmitted (the first time that STAFProc is started after the install) to a system within the IBM network. If you register information from a system that is outside the IBM network, the transmission will fail and the registration information will be discarded (note that this will not impact your ability to use STAF).

3.6. Advanced Options

Here is a list of the advanced options that are available during the install:

- Update Enviroment/Menus For:

  This allows you to specify the scope of environment/menu updates. The possible selections are:
  - System - the System environment variables and menus (if applicable) will be updated
  - User - the currently logged-in user's environment variables and menus (if applicable) will be updated
  - None - no environment variables or menus will be updated

  Note that on Unix systems, after the install completes, you must logout and log back in the have the system/user environment variables refreshed.

- Start STAF on user login (Windows only)

  This indicates whether you want STAF to automatically start when you login (by adding STAF to your Windows StartUp folder). This option is not available if you selected None for Update Environment/Menus. This option is only available with the InstallAnywhere installers.

- Create Start menu icons (Windows only)

  This indicates whether you want icons placed on the Windows "Start Programs" menu. This option is not available if you selected None for Update Environment/Menus. This option is only available with the InstallAnywhere installers.

- Start STAFProc (Windows only)

  This indicates whether you want STAFProc to be started in normal state (visible on the desktop) or minimized when the "Start STAF" icon in the Windows "Start Programs" menu is selected, or when STAFProc automatically starts when you login to Windows. This option is not available if you selected None for Update Environment/Menus. This option is only available with the InstallAnywhere installers.

- Default TCP/IP version

This indicates whether you want to use the IPv4 TCP libraries only, or if you want to use the TCP libraries that support both IPv4 and IPv6.

- Default Perl version (if applicable)

  This indicates which version of Perl you want to be the default Perl support for STAF.

- Use Perl version in System Path (if applicable)

  This indicates to determine the version of Perl in the system PATH at install-time (if possible) and use that version of Perl by default. If there is no version of Perl in the system PATH (or if the output of "perl -v" returns an unexpected value, i.e. something other than "5.6.x", "5.8.x", "5.10.x", "5.12.x", or "5.14.x") then the default version of Perl (determined by the "Default Perl version" option) will be used.

- Default Python version (if applicable)

  This indicates which version of Python you want to be the default Python support for STAF.

- Use Python version in System Path (if applicable)

  This indicates to determine the version of Python in the system PATH at install-time (if possible) and use that version of Python by default. If there is no version of Python in the system PATH (or if the output of "python -V" returns an unexpected value) then the default version of Python (determined by the "Default Python version" option) will be used.

- Default STAF Instance Name

  This allows you to specify the value for the STAF instance name for this version of STAF. Since multiple instances of STAFProc can be run at the same time on the same system, the STAF_INSTANCE_NAME environment variable can be used to specify a name for each STAFProc instance. You can leave the default instance name as "STAF", or you can change the instance name. Note, this value will only be used in the STAFEnv script file that will be created for you. This option is only available with the InstallAnywhere installers.

3.7. STAFEnv script

After the STAF install is complete, a STAFEnv.bat file (on Windows) or a STAFEnv.sh file (on Unix) will be created in the root STAF install directory. The STAFEnv script files are useful if you are going to be running two versions of STAF on the same machine and need a convenient way to switch settings for each version of STAF. An optional argument specifying the STAF instance name can be passed to a STAFEnv script file.

Here is a sample STAFEnv.bat file from a Windows system:

```
@echo off
REM STAF environment variables
set PATH=C:\STAF\bin;%PATH%
set CLASSPATH=C:\STAF\bin\JSTAF.jar;C:\STAF\samples\demo\STAFDemo.jar;%CLASSPATH%
set STAFCONVDIR=C:\STAF\codepage
if "%1" EQU "" set STAF_INSTANCE_NAME=STAF
if "%1" NEQ "" set STAF_INSTANCE_NAME=%1
```

Here is a sample STAFEnv.sh file from a Linux system:

```
#!/bin/sh
# STAF environment variables
PATH=/usr/local/staf/bin:${PATH:-}
LD_LIBRARY_PATH=/usr/local/staf/lib:${LD_LIBRARY_PATH:-}
CLASSPATH=/usr/local/staf/lib/JSTAF.jar:/usr/local/staf/samples/demo/STAFDemo.jar:${CLASSPATH:-}
STAFCONVDIR=/usr/local/staf/codepage
if [ $# = 0 ]
then
    STAF_INSTANCE_NAME=STAF
else
    if [ $1 != "start" ]
    then
        STAF_INSTANCE_NAME=$1
    else
        # Ignore "start" STAF instance name
        STAF_INSTANCE_NAME=STAF
    fi
fi
export PATH LD_LIBRARY_PATH CLASSPATH STAFCONVDIR STAF_INSTANCE_NAME
```

Note that to correctly execute the STAFEnv.sh script on Unix, you must execute it in this format:

```
. ./STAFEnv.sh
```

Or, if the STAFEnv.sh script is not in the current directory:

```
. ./usr/local/staf/STAFEnv.sh
```

Note that if STAFEnv.sh is executed during the bootup process on Unix, the bootup script may inadvertently pass the "start" argument to STAFEnv.sh, so when STAFProc is started during the system reboot, it will be using a STAF instance name of "start" instead of "STAF". To avoid this problem, the STAFEnv.sh script will ignore the STAF instance name argument if it is equal to "start".

3.8. startSTAFProc script

After the STAF install is complete, a startSTAFProc.bat file (on Windows) or a startSTAFProc.sh file (on Unix) will be created in the root STAF install directory. The startSTAFProc script files can be used to set up the STAF environment variables and start STAFProc.

Here is a sample startSTAFProc.bat file from a Windows system:

```
REM Sets up the STAF environment variables and starts STAFProc
call "C:\STAF\STAFEnv.bat"
start "Start STAF 3.3.5" /min "C:\STAF\bin\STAFProc.exe"
```

Here is a sample startSTAFProc.sh file from a Linux system:

```
#!/bin/sh
# Sets up the STAF environment variables and starts STAFProc
# in the background, logging STAFProc output to nohup.out
. /usr/local/staf/STAFEnv.sh
nohup /usr/local/staf/bin/STAFProc &
```

Note that to correctly execute the startSTAFProc.sh script on Unix, you must execute it in this format:

```
./startSTAFProc.sh
```

Or, if the startSTAFProc.sh script is not in the current directory:

```
/usr/local/staf/startSTAFProc.sh
```

4. Using InstallAnywhere to install STAF

4.1. IA Installer Arguments

Here are the command-line options for an IA installer (or uninstaller):

```
C:\>STAF3416-setup-win32.exe -help
    Usage: STAF3416-setup-win32 [-f <path_to_installer_properties_file> | -options]
          (to execute the installer)

  where options include:
    -?
        show this help text
    -i [swing | console | silent]
        specify the user interface mode for the installer
    -D<name>=<value>
        specify installer properties

  The options field may also include the following in case of uninstaller
  if it is enabled for Maintenance Mode
    -add <feature_name_1> [<feature_name_2 ...]
            Add Specified Features
    -remove <feature_name_1> [<feature_name_2 ...]
            Remove Specified Features
    -repair
            Repair Installation
    -uninstall
            Uninstall

  notes:
    1. the path to the installer properties file may be either absolute,
```

```
   or relative to the directory in which the installer resides.
2. if an installer properties file is specified and exists, all other
   command line options will be ignored.
3. if a properties file named either 'installer.properties' or
   <NameOfInstaller>.properties resides in the same directory as the
   installer, it will automatically be used, overriding all other command
   line options, unless the '-f' option is used to point to another valid
   properties file.
4. if an installer properties file is specified but does not exist, the
   default properties file, if present, will be used.  Otherwise, any
   supplied command line options will be used, or if no additional
   options were specified, the installer will be run using the default
   settings.
```

4.2. InstallAnywhere Graphical Install

The GUI (Graphical User Interfacae) mode is the default UI mode for a STAF InstallAnywhere installer. For InstallAnywhere to run in the GUI mode, it must be able to gain access to your X Windows display.

Note that if you start a graphical installation, but the system cannot gain access to the X Windows display (i.e. if you are accessing a Unix system via telnet or ssh), the installer will detect this situation and continue the installation in console mode. For example:

```
./STAF3416-setup-linux.bin
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...


Graphical installers are not supported by the VM. The console mode will be used
instead...

Preparing CONSOLE Mode Installation...
```

In this situation where you are accessing a Unix system via telnet or ssh, if you want to run in GUI mode instead of console mode and you have X Windows then you may be able to run in GUI mode if you set the DISPLAY environment variable to the workstation you are logged in to. For example:

```
export DISPLAY=yourhostname.company.com:0.0
```

replacing "yourhostname.company.com" with the host name or network address of the machine on which the display is running.

Here are examples of the screens that will be displayed in the InstallAnywhere Graphical Install. Note that these screen captures are from a Windows install; Unix installs will have similar screens.

Figure 1.

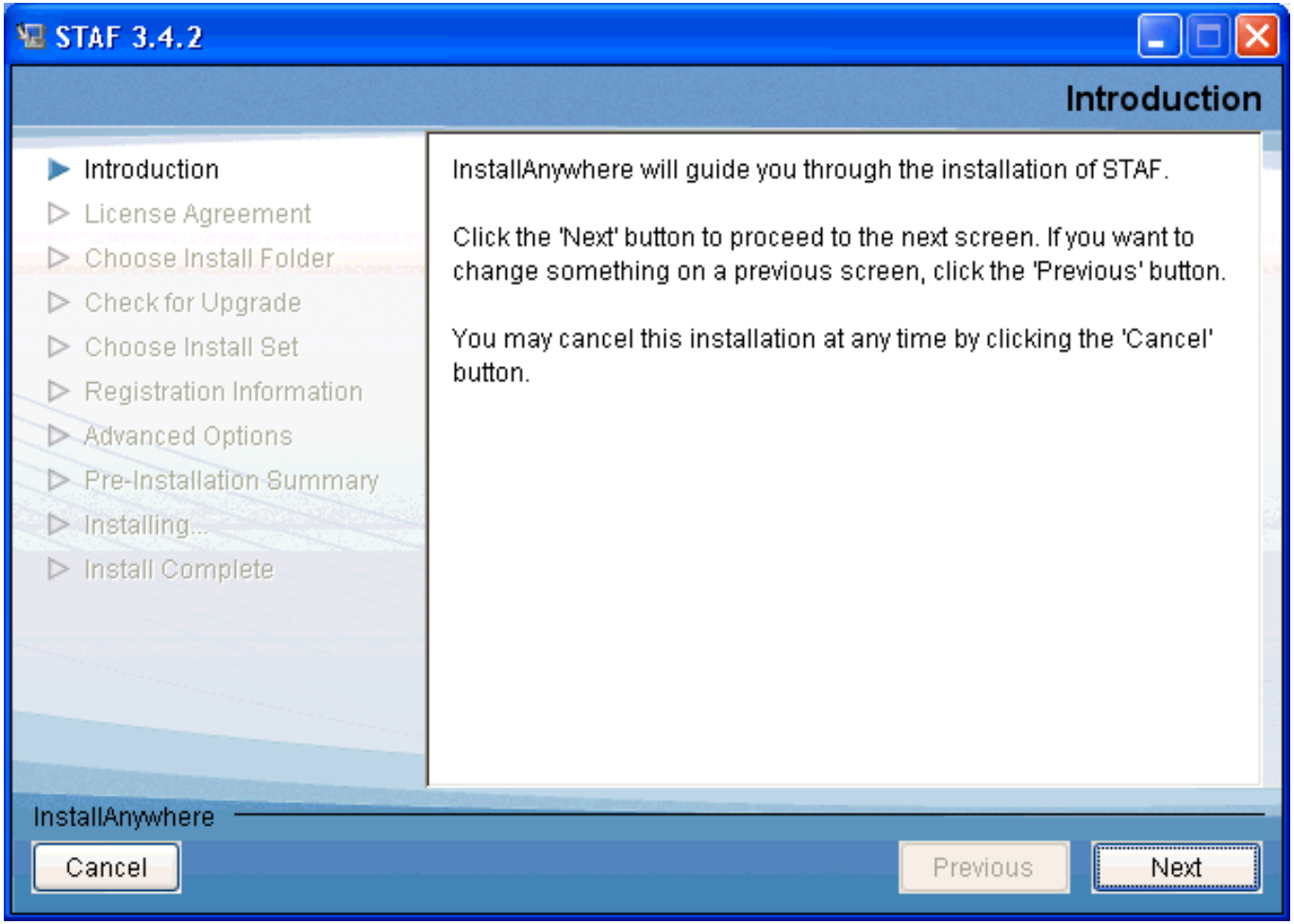


Figure 2.

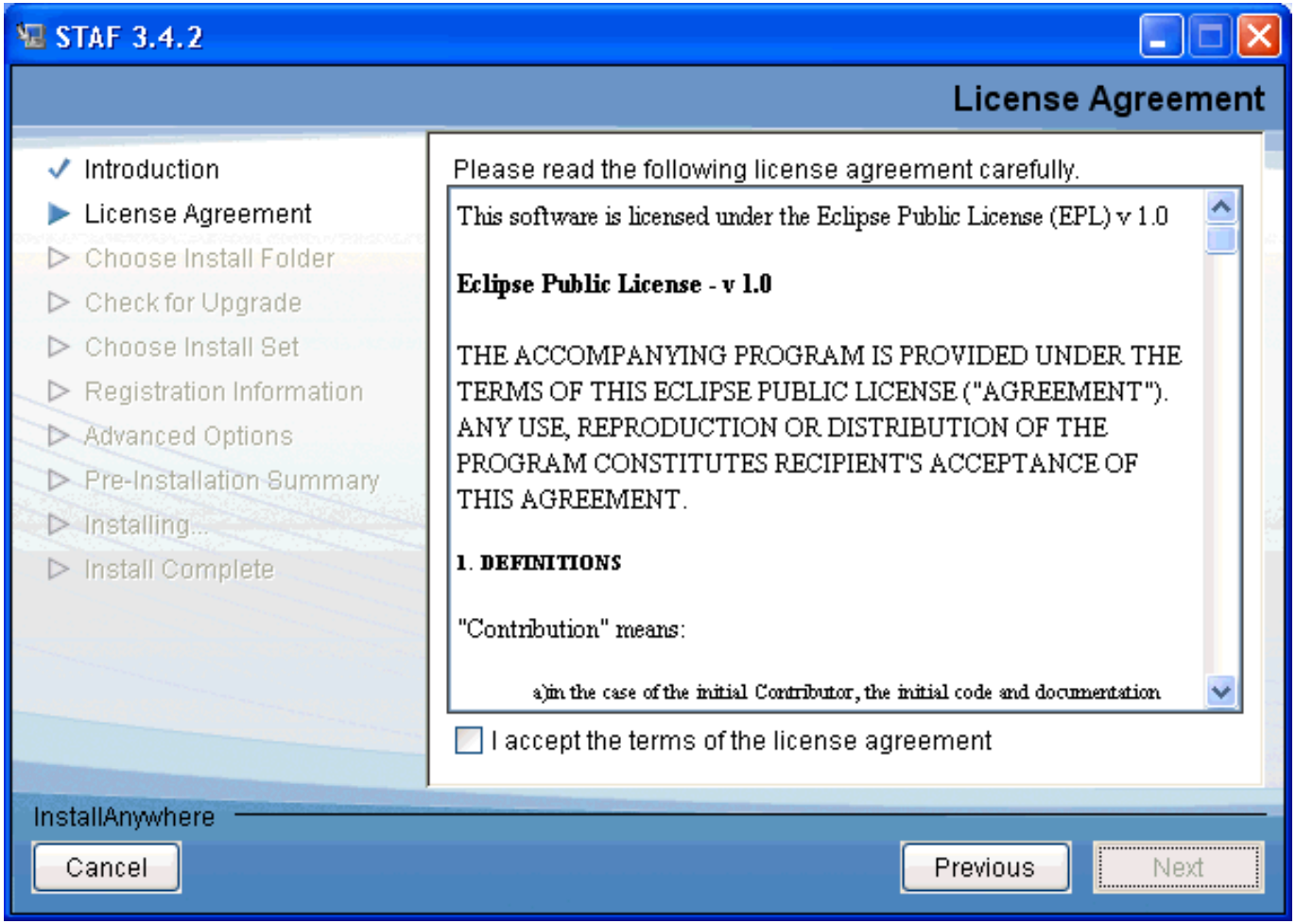

STAF 3.4.2
License Agreement
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Please read the following license agreement carefully.
This software is licensed under the Eclipse Public License (EPL) v 1.0
Eclipse Public License - v 1.0
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS
"Contribution" means:
a)in the case of the initial Contributor, the initial code and documentation
I accept the terms of the license agreement
InstallAnywhere
Cancel
Previous
Next

Figure 3.

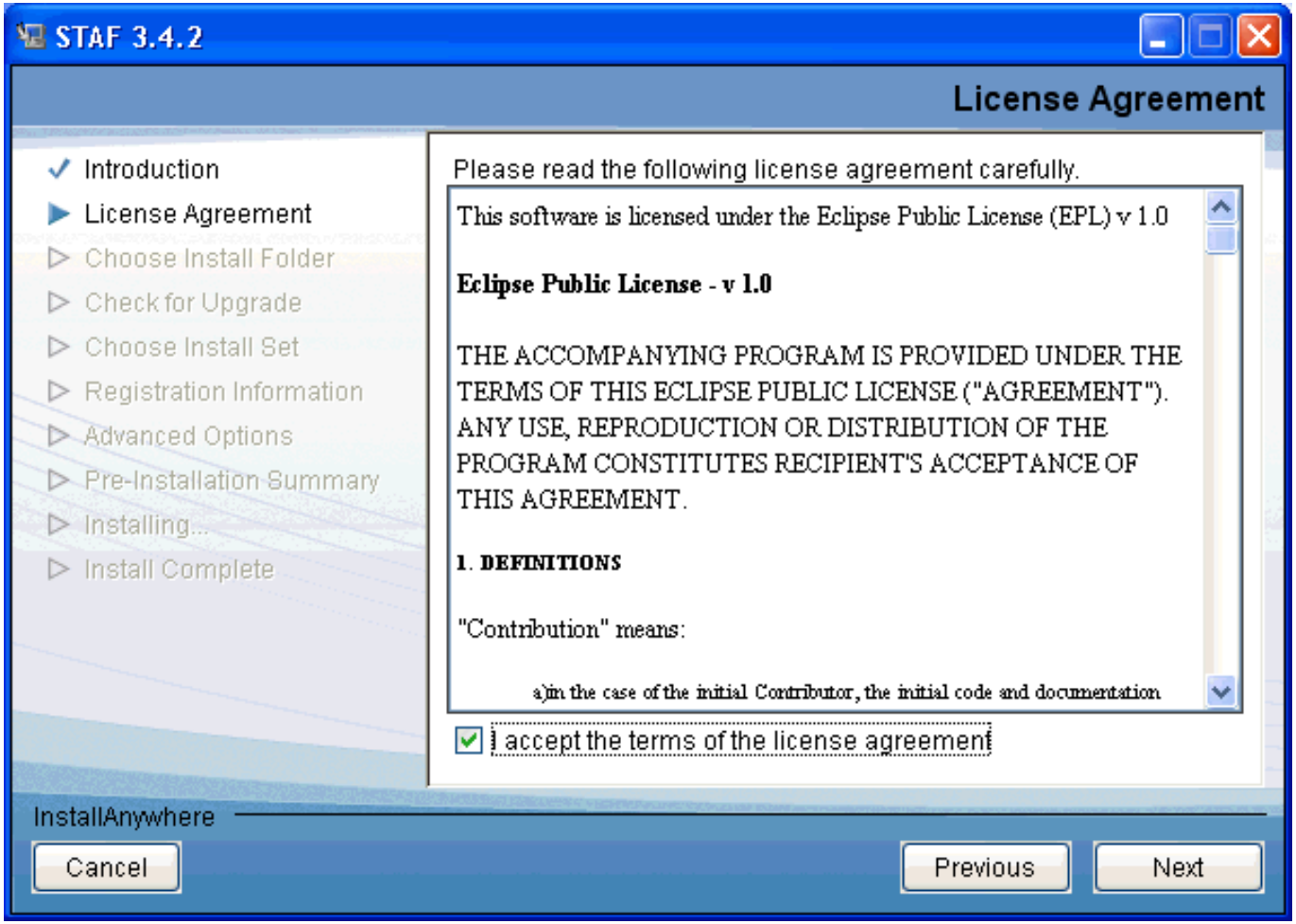

STAF 3.4.2
License Agreement
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Please read the following license agreement carefully.
This software is licensed under the Eclipse Public License (EPL) v 1.0
Eclipse Public License - v 1.0
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS
"Contribution" means:
a)in the case of the initial Contributor, the initial code and documentation
I accept the terms of the license agreement
InstallAnywhere
Cancel
Previous
Next

Figure 4.

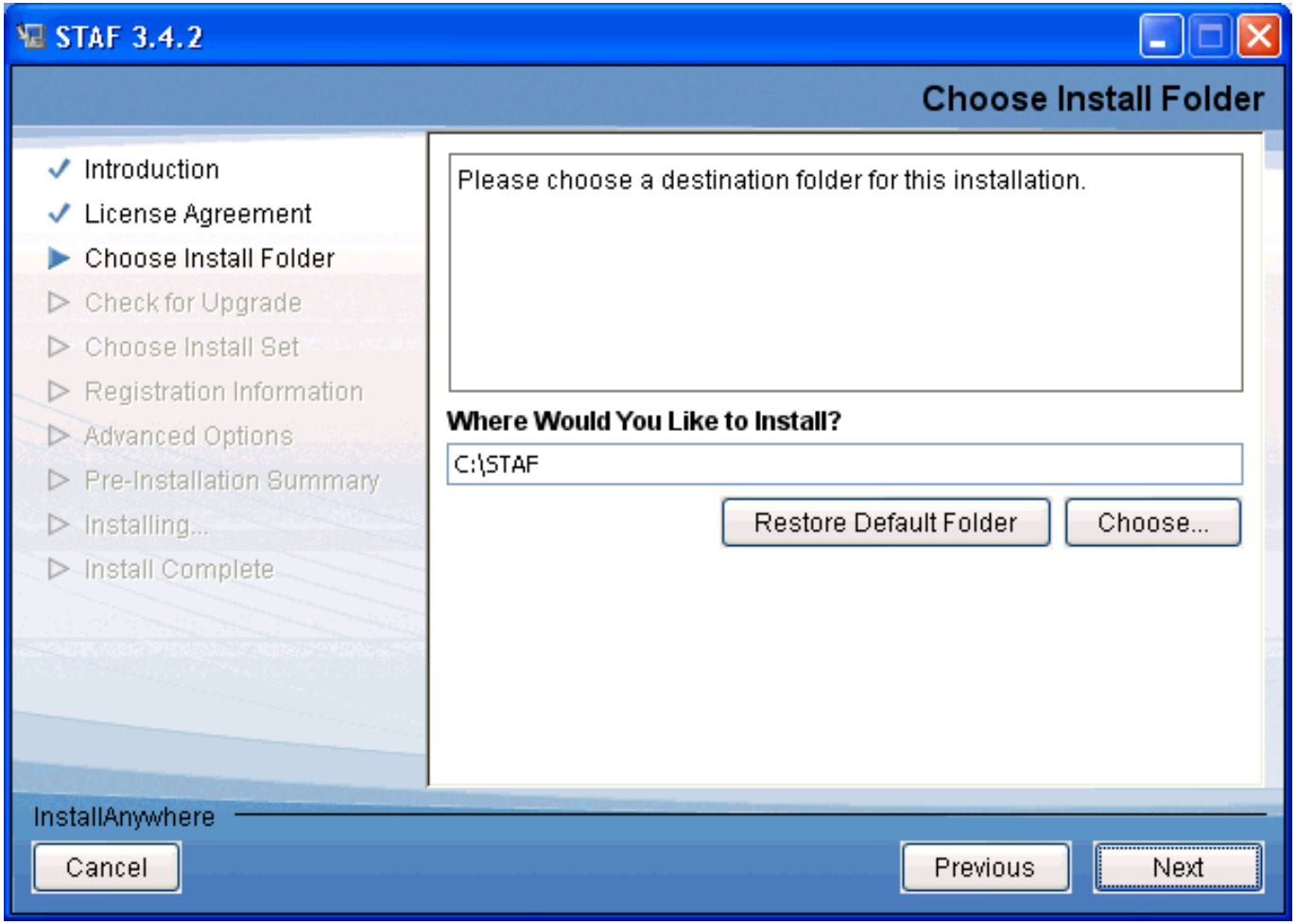

STAF 3.4.2
Choose Install Folder
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Please choose a destination folder for this installation.
Where Would You Like to Install?
C:\STAF
Restore Default Folder
Choose...
InstallAnywhere
Cancel
Previous
Next

Figure 5.

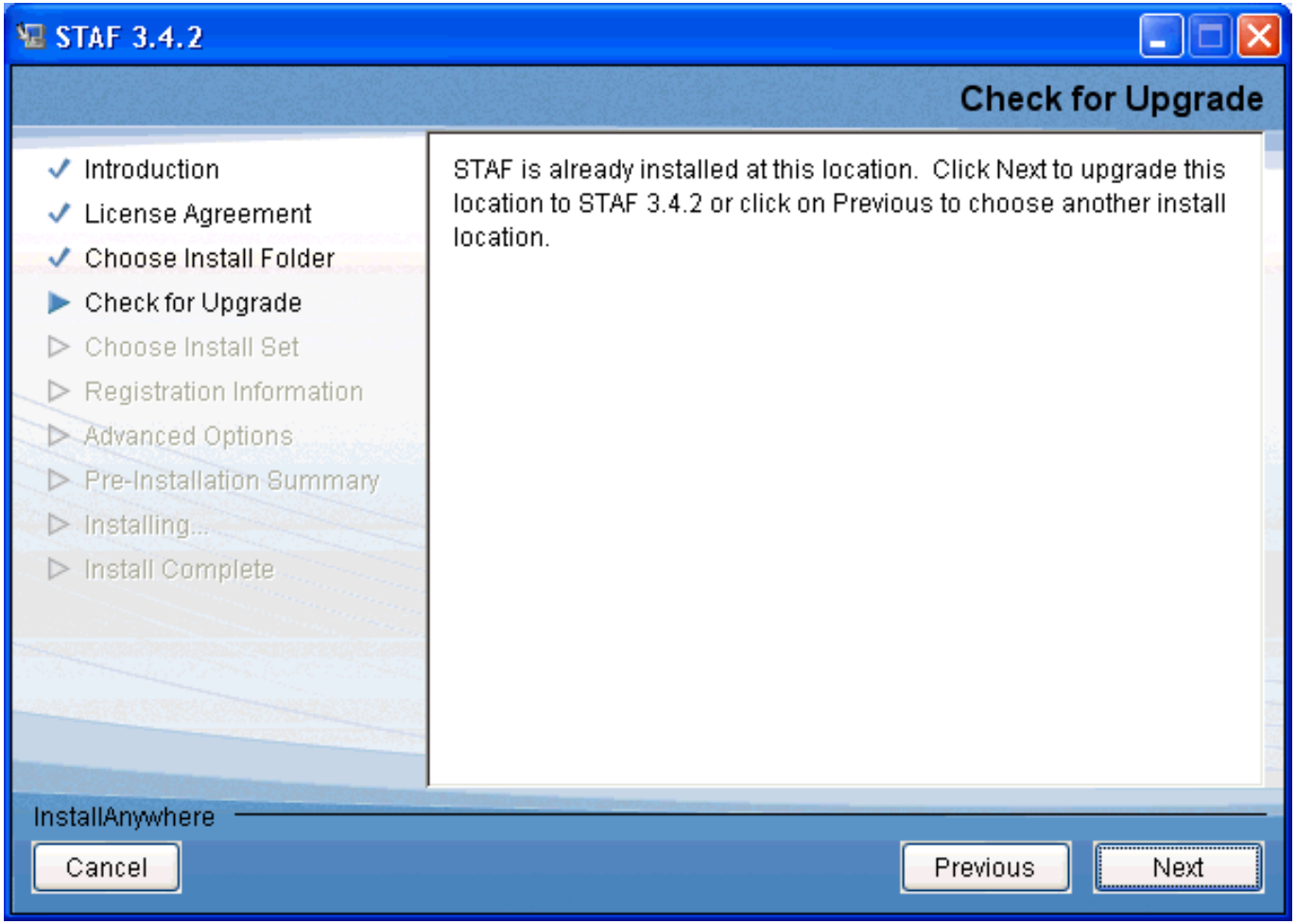

STAF 3.4.2
Check for Upgrade
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
STAF is already installed at this location. Click Next to upgrade this location to STAF 3.4.2 or click on Previous to choose another install location.
InstallAnywhere
Cancel
Previous
Next

Figure 6.

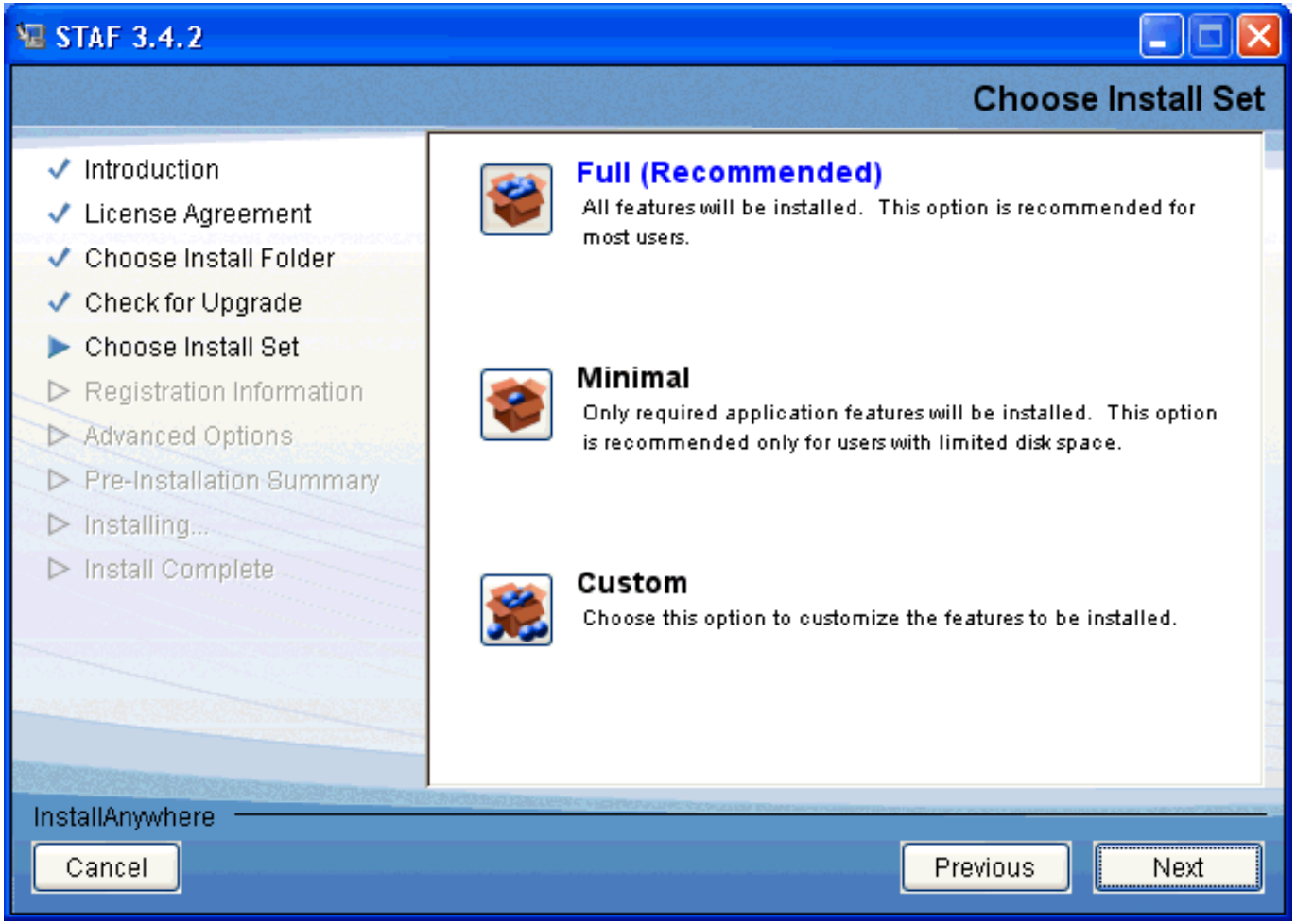

STAF 3.4.2
Choose Install Set
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Full (Recommended)
All features will be installed. This option is recommended for most users.
Minimal
Only required application features will be installed. This option is recommended only for users with limited disk space.
Custom
Choose this option to customize the features to be installed.
InstallAnywhere
Cancel
Previous
Next

Figure 7.

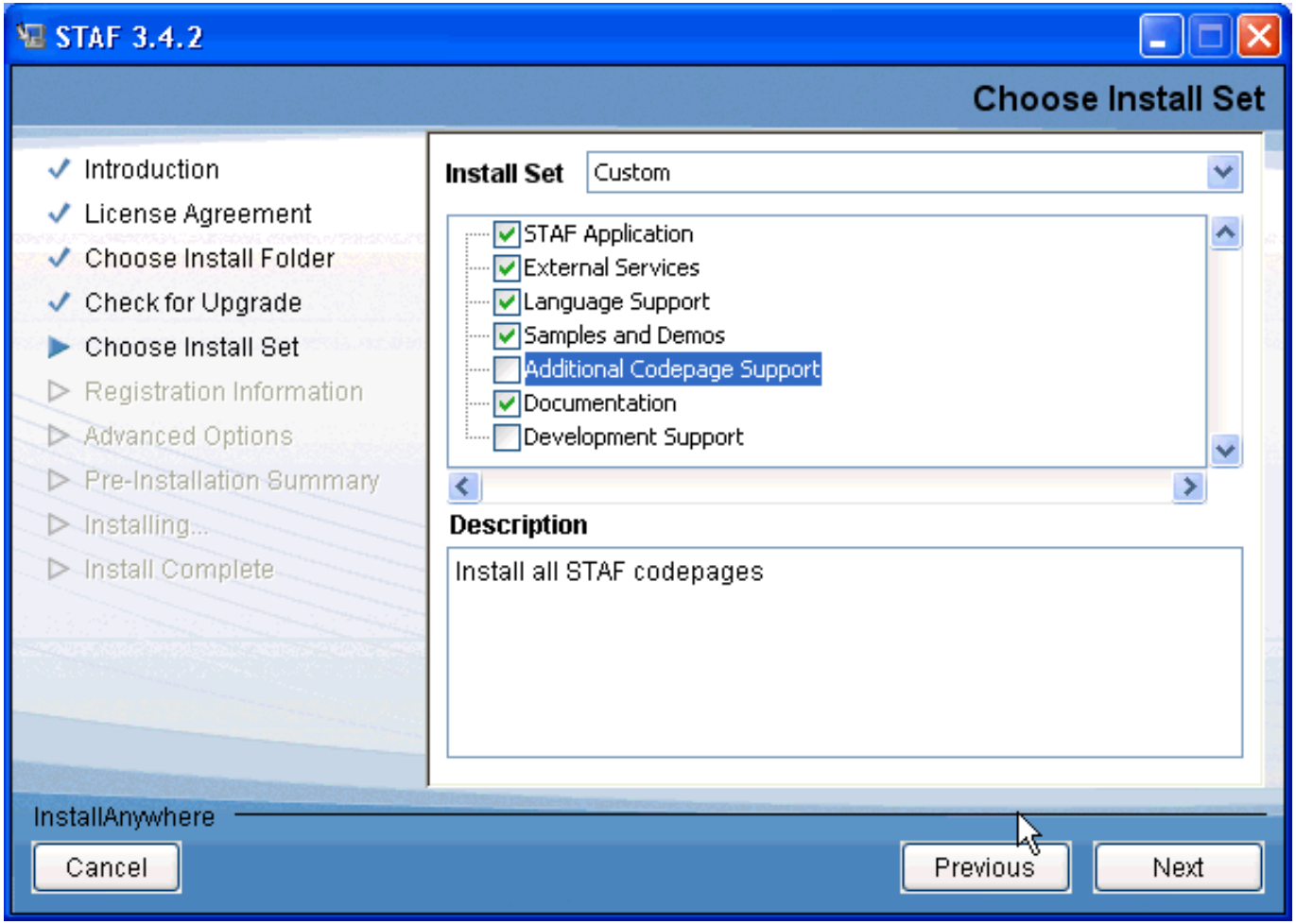

STAF 3.4.2
Choose Install Set
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Install Set
Custom
STAF Application
External Services
Language Support
Samples and Demos
Additional Codepage Support
Documentation
Development Support
Description
Install all STAF codepages
InstallAnywhere
Cancel
Previous
Next

Figure 8.

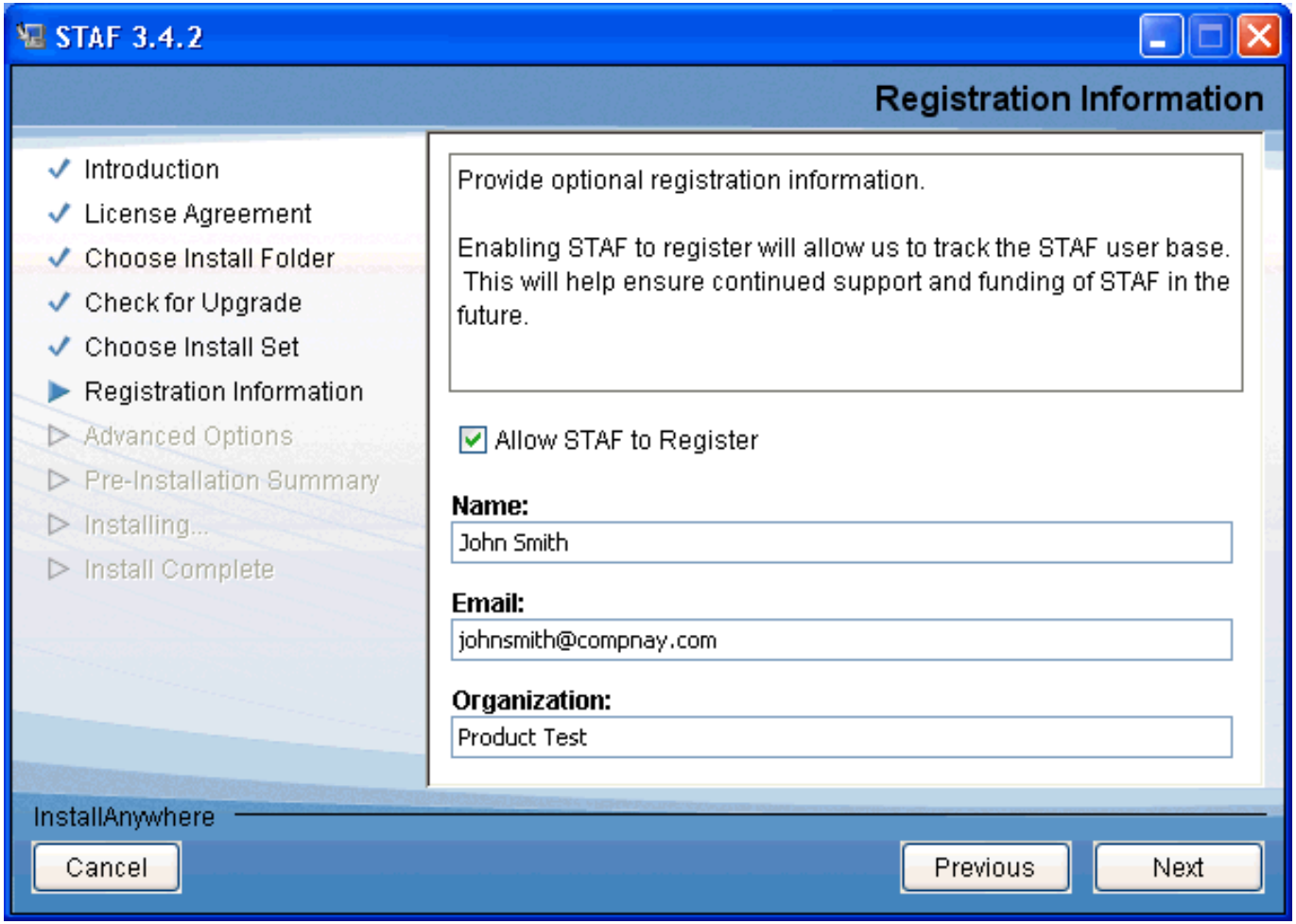

STAF 3.4.2
Registration Information
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Provide optional registration information.
Enabling STAF to register will allow us to track the STAF user base. This will help ensure continued support and funding of STAF in the future.
Allow STAF to Register
Name:
John Smith
Email:
johnsmith@compnay.com
Organization:
Product Test
InstallAnywhere
Cancel
Previous
Next

Figure 9.

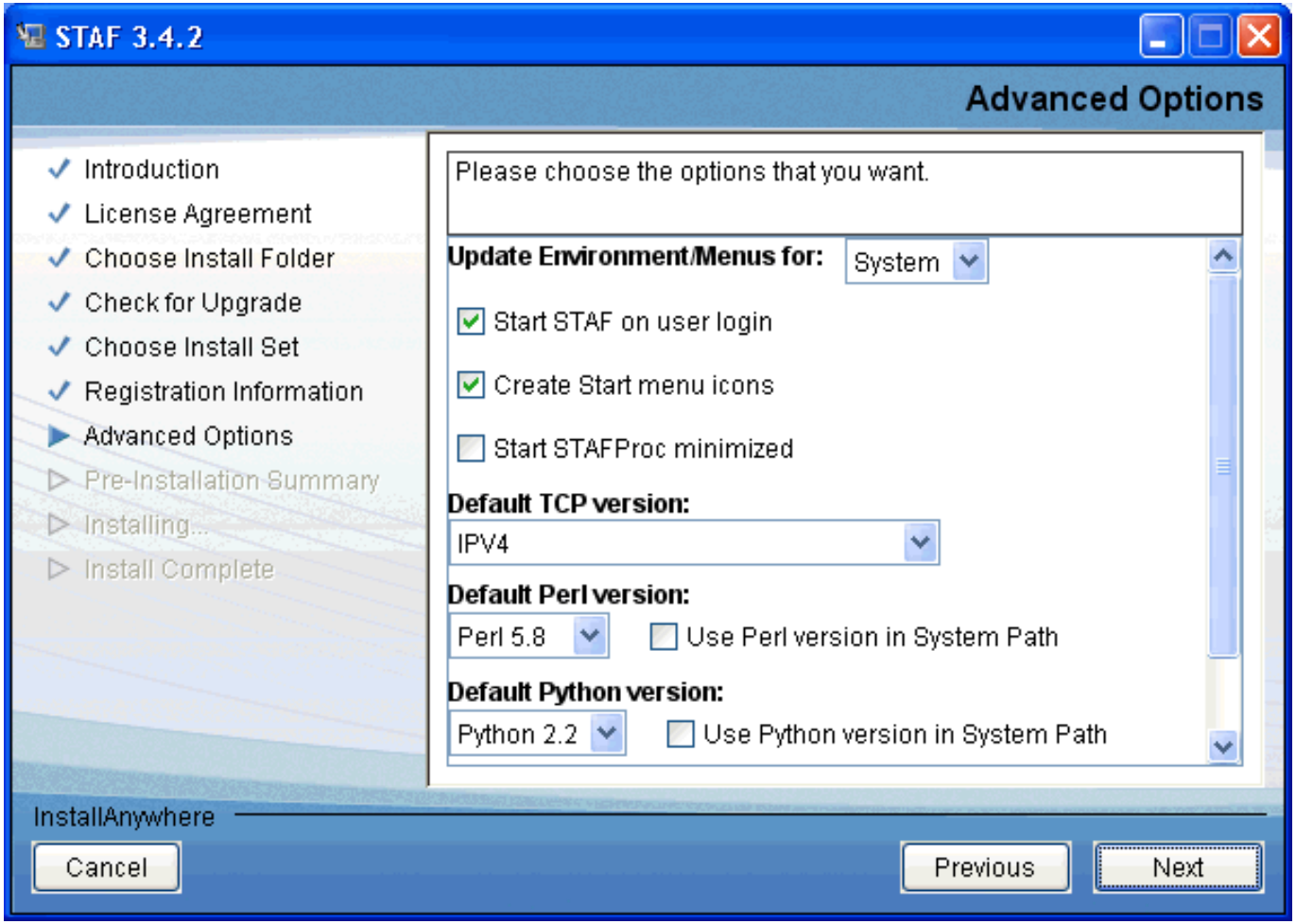

STAF 3.4.2
Advanced Options
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Please choose the options that you want.
Update Environment/Menus for:
System
Start STAF on user login
Create Start menu icons
Start STAFProc minimized
Default TCP version:
IPV4
Default Perl version:
Perl 5.8
Use Perl version in System Path
Default Python version:
Python 2.2
Use Python version in System Path
InstallAnywhere
Cancel
Previous
Next

Figure 10.

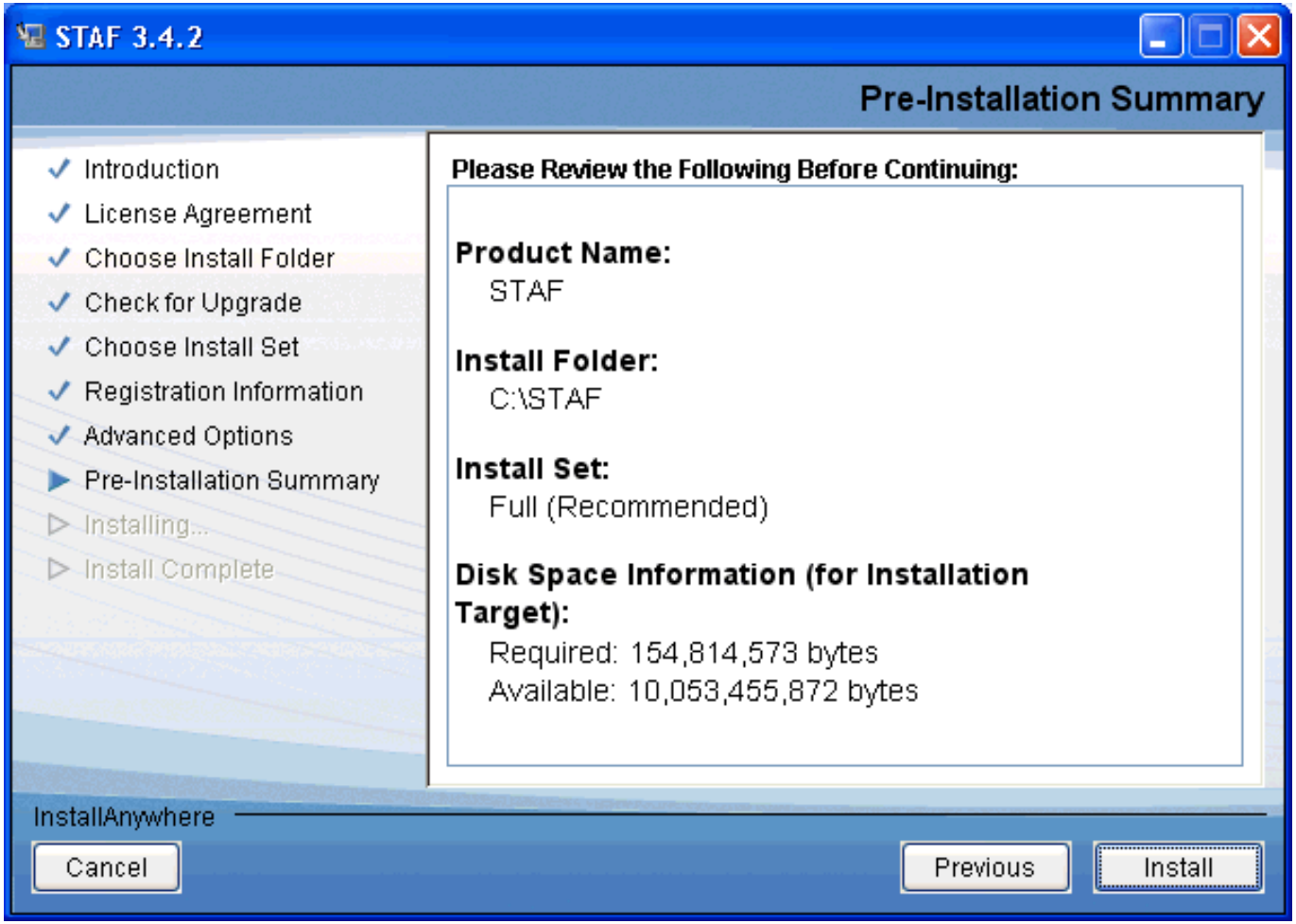

STAF 3.4.2
Pre-Installation Summary
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Please Review the Following Before Continuing:
Product Name:
STAF
Install Folder:
C:\STAF
Install Set:
Full (Recommended)
Disk Space Information (for Installation Target):
Required: 154,814,573 bytes
Available: 10,053,455,872 bytes
InstallAnywhere
Cancel
Previous
Install

Figure 11.

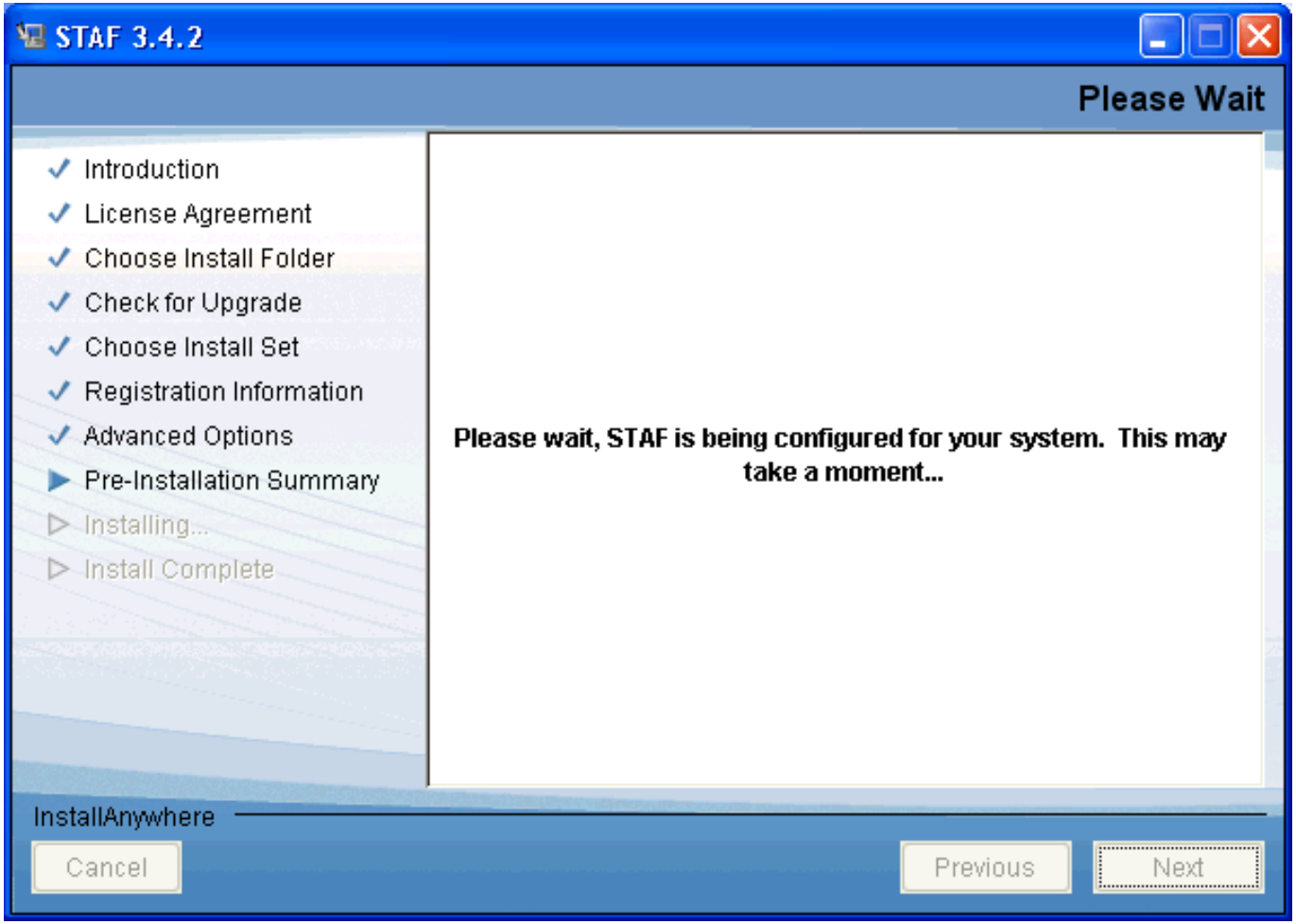

STAF 3.4.2
Please Wait
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Please wait, STAF is being configured for your system. This may take a moment...
InstallAnywhere
Cancel
Previous
Next

Figure 12.

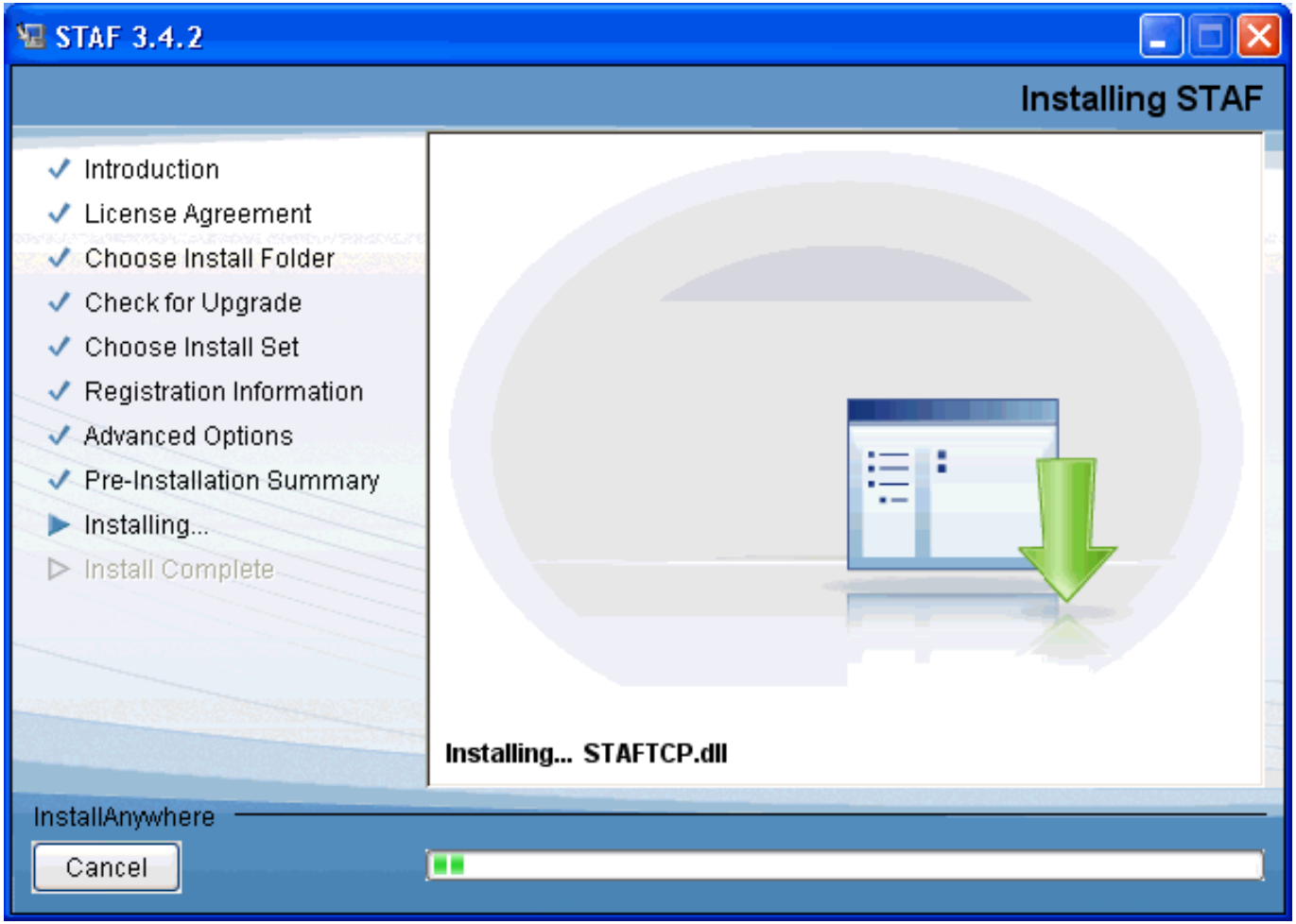
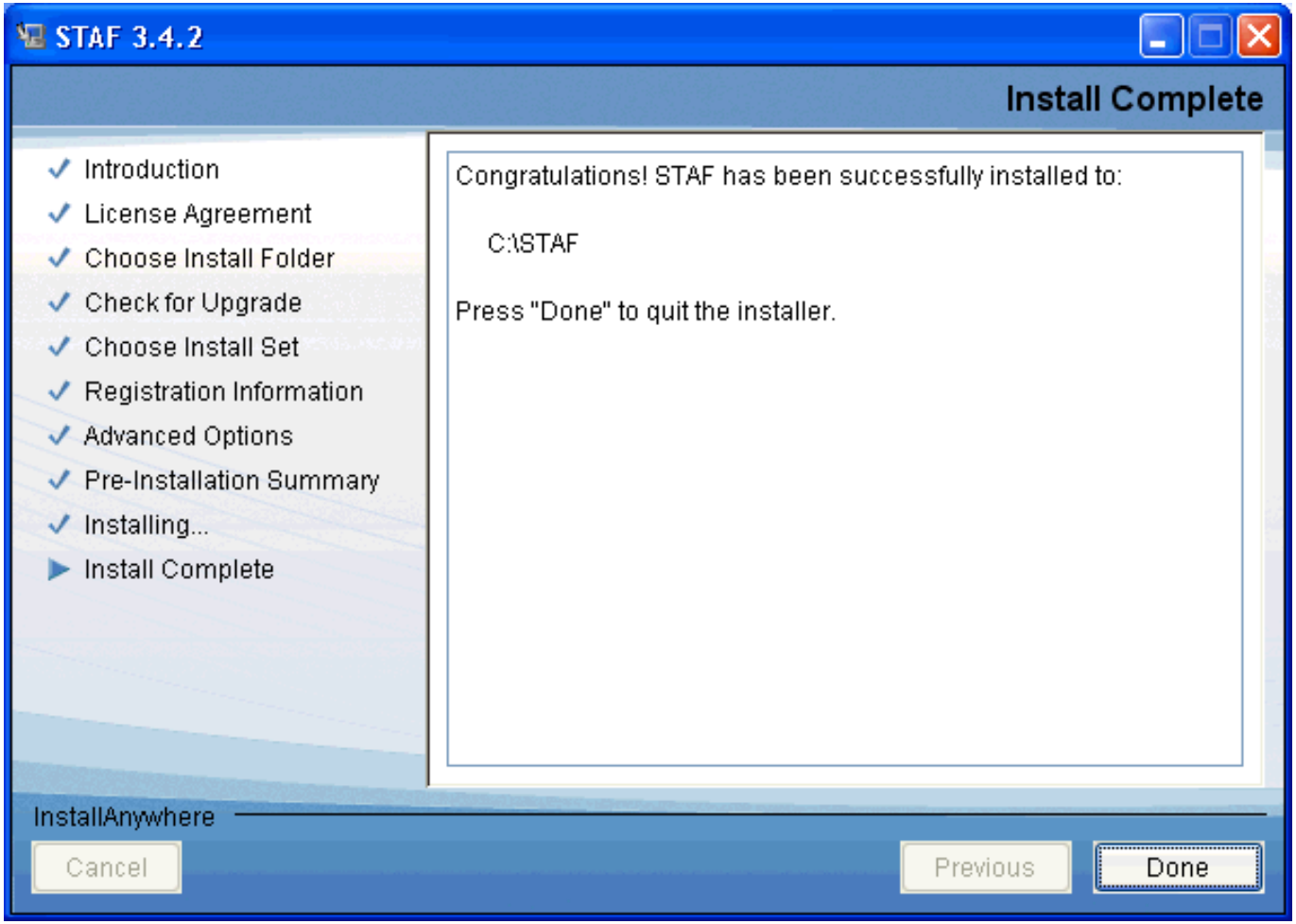

STAF 3.4.2
Installing STAF
Introduction
License Agreement
Choose Install Folder
Check for Upgrade
Choose Install Set
Registration Information
Advanced Options
Pre-Installation Summary
Installing...
Install Complete
Installing... STAFTCP.dll
InstallAnywhere
Cancel

Figure 13.



4.3. InstallAnywhere Silent Install

To start in IA install in silent mode, you can specify the "-i silent" property when starting the installer. You can also use specify this property in a response file.

The following table shows all of the install properties that are available for a silent installation. Note that these options can be specified at the command line or in a response file.

Property Name	Description	Value	Equivalent ISMP option

ACCEPT_LICENSE	Acceptance of the license agreement.	Values are "1" (accept) or "0" (or any value other than "1"). Default is "0"	license.selection
USER_INSTALL_DIR	The directory where STAF will be installed.	Default is "C:\STAF" on Windows, "/Library/staf" on Mac OS X, and "/usr/local/staf" on other Unix platforms.	stafinstalldirectory.defaultInstallLocation
INSTALLER_UI	The install UI type.	Values are "silent", "swing", or "console". This allows you to specify the install UI type without using the "-i" option. Default is "swing".	None
CHOSEN_INSTALL_SET	The install set.	Values are "Full", "Minimal", or "Custom". Default is "Full".	setupTypes.selectedSetupTypeId
CHOSEN_INSTALL_FEATURE_LIST	The Custom features to install.	Valid features names are: STAF, ExtSvcs,Langs, Samples,Codepage, Docs,Develop. The features should be specified as a comma-separated list of the feature names. Note that this option has no effect for "Full" or "Minimal" install.	None

REGISTER	Whether to perform STAF registration when STAF is first started.	Values are "1" or "0". Default is "1".	None
REGISTRATION_NAME	The name to register.	Default is "".	None
REGISTRATION_EMAIL	The email to register.	Default is "".	None
REGISTRATION_ORG	The organization to register.	Default is "".	None
UPDATE_ENVIRONMENT	Scope of environment/menu updates.	Values are "System", "User", "None". Default is "System".	stafOptions.updateEnvironmentVariables
START_ON_LOGIN	Whether to start STAF on Login. This is a Windows-only property.	Values are "1" or "0". Default is "1".	None
CREATE_START_MENU_ICONS	Whether to create Start menu icons/ programs folder for STAF. This is a Windows-only property.	Values are "1" or "0". Default is "1".	None
START_STAFPROC	Whether to start the STAFProc window in normal state (visible on the desktop) or minimized. This is a Windows-only property.	Values are "Normal" or "Minimized". Default is "Normal".	None
USE_TCP_VERSION	The default TCP/IP version.	Values are "IPV4" or "IPV4_IPV6". "IPV4" is the default.	stafOptions.defaultIPvVersion
USE_PERL_VERSION	The default Perl version. Only valid on platforms with Perl support.	Values are "5.8", "5.10", "5.12", "5.14", and "5.6" (note that the supported Perl versions will vary depending on the operating system). Default is "5.8".	None

USE_PERL_SYSTEM_PATH	Whether to determine the version of Perl in the system PATH at install-time (if possible) and use that version of Perl by default. Only valid on platforms with Perl support. If there is no version of Perl in the system PATH (or if the output of "perl -v" returns an unexpected value, i.e. something other than "5.6.x", "5.8.x", "5.10.x", "5.12.x", or "5.14.x") then the default version of Perl (determined by the USE_PERL_VERSION option) will be used.	Values are "0", "1". Default is "0".	None
USE_PYTHON_VERSION	The default Python version. Only valid on platforms with Python support.	Values are "2.2", "2.3", "2.4", "2.5", "2.6", "2.7", "3.0", and "3.1" (note that the supported Python versions will vary depending on the operating system). Default is "2.2" except on Mac OS X (default is "2.3"), FreeBSD (default is "2.4"), and Windows AMD64 (default is "2.5").	None

USE_PYTHON_SYSTEM_PATH	Whether to determine the version of Python in the system PATH at install-time (if possible) and use that version of Python by default. Only valid on platforms with Python support. If there is no version of Python in the system PATH (or if the output of "python -V" returns an unexpected value, i.e. something other than "2.2.x", "2.3.x", "2.4.x", "2.5.x", "2.6.x", "2.7.x", "3.0", or "3.1") then the default version of Python (determined by the USE_PYTHON_VERSION option) will be used.	Values are "0", "1". Default is "0".	None
USE_TCL_VERSION	The default TCL version. Only valid on platforms with TCL support.	Values are "8.3", "8.4", "8.5" and "8.6" (note that the supported TCL versions will vary depending on the operating system). Default is "8.3", or "8.4" on platforms that do not support TCL 8.3, or "8.5" on platforms that do not support TCL 8.4.	None
USE_TCL_SYSTEM_PATH	Whether to determine the version of TCL in the system PATH at install-time (if possible) and use that version of TCL by default. Only valid on platforms with TCL support. If there is no version of TCL in the system PATH then the default version of TCL (determined by the USE_TCL_VERSION option) will be used.	Values are "0", "1". Default is "0".	None
STAF_INSTANCE_NAME	The default STAF instance name.	Default is "STAF".	None

USER_REQUESTED_RESTART	Whether to automatically restart the Windows operating system if target files are in use during the STAF installation.	Values are "YES" and "NO". Default is "NO".	None

Note that the property names and values are case-sensitive.

Here are some examples of specifying silent install properties:

- Perform a silent install with all default options:

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1
```

- Install STAF in C:\Program Files\STAF:

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DUSER_INSTALL_DIR="C:\Program Files\STAF"
```

- Perform a silent install of STAF, where the STAFProc window will start minimized:

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DSTART_STAFPROC=Minimized
```

- Perform a Minimal install of STAF, without updating any menus or environment variables, and using the IPV4/IPv6-enabled TCP libraries:

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DCHOSEN_INSTALL_SET=Minimal
    -DUPDATE_ENVIRONMENT=None -DUSE_TCP_VERSION=IPV4_IPV6
```

- Perform a Custom install of STAF in directory C:\tools\staf, with features STAF, External Services, Languages, and all codepages:

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DUSER_INSTALL_DIR=C:\tools\staf
    -DCHOSEN_INSTALL_SET=Custom -DCHOSEN_INSTALL_FEATURE_LIST=STAF,ExtSvcs,Langs,Codepage
```

- Perform a silent install of STAF, determining the version of Perl in the system PATH at install-time (if possible) and using that version of Perl by default. If there is no version of Perl in the system PATH at install-time, then use Perl 5.10 by default.

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DUSE_PERL_SYSTEM_PATH=1
    -DUSE_PERL_VERSION=5.10
```

- Perform a silent install of STAF, determining the version of Python in the system PATH at install-time (if possible) and using that version of

Python by default. If there is no version of Python in the system PATH at install-time, then use Python 2.6 by default.

```
STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DUSE_PYTHON_SYSTEM_PATH=1
    -DUSE_PYTHON_VERSION=2.6
```

You can specify a properties (response) file to use when starting the STAF IA Installer. The properties file should contain the same -D options that you would specify on a command-line installation. For example:

```
STAF3416-setup-win32 -f installer.properties
```

Here is a sample installer.properties file:

```
INSTALLER_UI=silent
ACCEPT_LICENSE=1
USER_INSTALL_DIR=C:\\STAF
CHOSEN_INSTALL_SET=Full
REGISTER=1
REGISTRATION_NAME=Joe Smith
REGISTRATION_EMAIL=address@comany.com
REGISTRATION_ORG=ABC Test
UPDATE_ENVIRONMENT=System
START_ON_LOGIN=1
CREATE_START_MENU_ICONS=1
START_STAFPROC=Minimized
USE_TCP_VERSION=IPV4
USE_PERL_VERSION=5.8
USE_PERL_SYSTEM_PATH=1
USE_PYTHON_VERSION=2.2
USE_PYTHON_SYSTEM_PATH=1
STAF_INSTANCE_NAME=STAF
USER_REQUESTED_RESTART=NO
```

To perform a GUI install and have the selected options recorded in a properties file, you can specify the "-r file-path" option when starting the IA installer. For example:

```
STAF3416-setup-win32.exe -r c:/temp/installer.properties
```

This will create a "c:/temp/installer.properties" file. Note that you will need to manually add the *INSTALLER_UI=<mode>* line.

4.4. InstallAnywhere Console Install

In IA to perform a console installation, use the "-i console" option. For example:

```
./STAF3416-setup-linux.bin -i console
```

Here are examples (on Linux) of the console panels that are displayed:

Figure 14.

```
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

Preparing CONSOLE Mode Installation...

===============================================================================
STAF                                               (created with InstallAnywhere)
-------------------------------------------------------------------------------




===============================================================================
Introduction
------------

InstallAnywhere will guide you through the installation of STAF.

It is strongly recommended that you quit all programs before continuing with
this installation.

Respond to each prompt to proceed to the next step in the installation.  If you
want to change something on a previous step, type 'back'.

You may cancel this installation at any time by typing 'quit'.
```

```
PRESS <ENTER> TO CONTINUE:
```

Figure 15.

```
================================================================================
License Agreement
-----------------

Installation and Use of STAF Requires Acceptance of the Following License
Agreement:

This software is licensed under the Eclipse Public License (EPL) v 1.0

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation
distributed under this Agreement, andb) in the case of each subsequent
Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a
Contributor if it was added to the Program by such Contributor itself or anyone

PRESS <ENTER> TO CONTINUE:

  ...  ...  ...  ...  ...  ...  ...  ...  ...

DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT? (Y/N): Y
```

Figure 16.

```
================================================================================
Choose Install Folder
---------------------

Where would you like to install?

  Default Install Folder: /usr/local/staf

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
      :
```

Figure 17.

```
================================================================================
Choose Install Set
------------------

Please choose the Install Set to be installed by this installer.

  ->1- Full (Recommended)
    2- Minimal

    3- Customize...

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   :
```

Figure 18.

```
================================================================================
Registration Information
------------------------

Allow STAF to Register? (DEFAULT: 1):
Registration Name: Jane Smith
Registration Email: janesmith@company.com
Registration Organization: Product Test
```

Figure 19.

```
================================================================================
Advanced Options
----------------

  ->1- System
    2- User
    3- None

Update Environment/Menus for:


  ->1- IPV4
    2- IPV4_IPV6

Default TCP version:


  ->1- 5.8
    2- 5.10
    3- 5.12
    4- 5.14
    5- 5.6

Default Perl version:


Use Perl version in System Path? (DEFAULT: 0):


  ->1- 2.2
    2- 2.3
    3- 2.4
    4- 2.5
    5- 2.6
    6- 2.7
    7- 3.0
```

```
    8- 3.1

Default Python version:


Use Python version in System Path? (DEFAULT: 0):


  ->1- 8.4
    2- 8.5
    3- 8.6

Default TCL version:


Use TCL version in System Path? (DEFAULT: 0):


Default STAF Intance Name (DEFAULT: STAF):
```

Figure 20.

```
===============================================================================
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Product Name:
    STAF

Install Folder:
    /usr/local/staf

Install Set
    Full (Recommended)

Disk Space Information (for Installation Target):
    Required:  91,109,690 bytes
```

```
    Available: 16,505,991,168 bytes

PRESS <ENTER> TO CONTINUE:
```

Figure 21.

```
================================================================================
Installing...
-------------

 [==================|==================|==================|==================]
 [------------------|------------------|------------------|------------------]
```

Figure 22.

```
================================================================================
Installation Complete
---------------------

Congratulations. STAF has been successfully installed to:

   /usr/local/staf

PRESS <ENTER> TO EXIT THE INSTALLER:
```

4.5. Installer Return Codes

The STAF IA installers will return the following return codes:

- 0 - if the install was successful
- 1 - if there were any errors during the install
- 200 - if a silent install was performed and the -DACCEPT_LICENSE=1 option was not specified

4.6. Install Log

The STAF IA installers will create a log file called STAFInstall.log in the root install directory.

Note that the log file will always be created (it is not an option that can be specified when starting the installer) and will be overwritten on each install.

The install log will contain a "Summary" section with status information for the install.

For example, the Summary for a successful install would look like:

```
Summary
-------

Installation: Successful.

357 Successes
0 Warnings
0 NonFatalErrors
0 FatalErrors
```

If there were any errors during the install, the Summary would look like:

```
Summary
-------

Installation: Successfulwith errors.

287 Successes
0 Warnings
3 NonFatalErrors
0 FatalErrors
```

If there were any errors during the install, the "Install Log Detail" section will contain information about the errors. For example:

```
Install File:             /usr/local/staf/lib/libJSTAF.sl
                          Status: ERROR
                          Additional Notes: ERROR - ZeroGs1: /usr/local/staf/lib/libJSTAF.sl (Text
file busy (errno:26))

Install File:             /usr/local/staf/lib/libJSTAFSH.sl
```

```
                          Status: ERROR
                          Additional Notes: ERROR - ZeroGs1: /usr/local/staf/lib/libJSTAFSH.sl
(Text file busy (errno:26))

Install File:             /usr/local/staf/lib/libSTAFLIPC.sl
                          Status: ERROR
                          Additional Notes: ERROR - ZeroGs1: /usr/local/staf/lib/libSTAFLIPC.sl
(Text file busy (errno:26))
```

4.7. Specifying a Temporary Directory

In ISMP, you could override the default temporary directory used during the install (for example, if the default temporary directory did not contain enough free space) by using the "is:tempdir <dir>" option.

In IA, on Windows, the installer uses the directory in the TEMP environment variable, and on Unix, the installer uses the directory in the IATEMPDIR environment variable. So, prior to running the installer, set the appropriate environment variable to a directory containing enough free space.

4.8. Uninstaller

The IA uninstaller for STAF is created in directory <install-root>/Uninstall_STAF. The uninstaller executable is <install-root>/Uninstall_STAF/Uninstall_STAF.exe on Windows and <install-root>/Uninstall_STAF/Uninstall_STAF on Unix.

Note that by default, the uninstaller will execute in the mode used for the installation. For example, if the install mode was silent, the uninstaller will run in silent mode by default.

4.9. Comparison of IA install options and ISMP install options

Here are some comparisons of silent install options for STAF 3.2.5 (using ISMP) and STAF 3.3.0 and later (using IA):

- Perform a silent install using all of the default options

```
(ISMP) STAF325-setup-win32 -is:log c:\temp\ismplog.txt -silent -W license.selection="Accept"

(IA)   STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1
```

- Perform a silent install to C:\tools\staf

```
(ISMP) STAF325-setup-win32 -is:log c:\temp\ismplog.txt -silent -W license.selection="Accept" -
W stafinstalldirectory.defaultInstallLocation="C:\tools\staf"
```

```
(IA)   STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DUSER_INSTALL_DIR=C:\tools\staf
```

- Perform a a silent installation of install set "Minimal", without updating environment variables/menus, and with the support for IPv4 and IPv6

```
(ISMP) STAF325-setup-win32 -is:log c:\temp\ismplog.txt -silent -W license.selection="Accept" -
W setupTypes.selectedSetupTypeId="Minimal" -W stafOptions.updateEnvironmentVariables="None" -W
stafOptions.defaultIPvVersion="IPv4 and IPv6"

(IA)   STAF3416-setup-win32 -i silent -DACCEPT_LICENSE=1 -DCHOSEN_INSTALL_SET=Minimal -
DUPDATE_ENVIRONMENT=None -DUSE_TCP_VERSION=IPV4_IPV6
```

- Perform a silent uninstall

```
(ISMP) ProductDir\_uninst\uninstaller -silent

(IA)   ProductDir\Uninstall_STAF\Uninstall_STAF.exe -i silent
```

5. Using STAFInst to install STAF

5.1. STAFInst Install

A single gzipped or compressed tar file is provided for Unix systems. For some platforms that are not supported by InstallAnywhere, such as z/OS and AS/400, this is the only install file provided. Before installing STAF, you will need to gunzip the gzipped tar file or uncompress the compressed tar file and then untar it into a temporary directory from which you will later run the STAFInst command. Note that when you untar the file, it will expand into a single directory named "staf" that will contain the installation files and directories.

For example, assuming you downloaded the gzipped tar file for Linux into a temporary directory named /tmp/STAF,

```
cd /tmp/STAF
gunzip STAF3416-linux.tar.gz
tar -xvf STAF3416-linux.tar
cd staf
```

Or, for example, assuming you downloaded the compressed tar file for z/OS into a temporary directory named /tmp/STAF,

```
cd /tmp/STAF
compress -d STAF3416-zos.tar.Z
tar -xvf STAF3416-zos.tar
cd staf
```

Note that you can delete this temporary directory (e.g. /tmp/STAF) after you have run the STAFInst command to install STAF to the target directory.

Optionally, you may prefer to untar STAF on one system and mount the installable images from other systems.

Once the STAF installation image is ready, you should run the STAFInst command. The syntax is as follows.

```
Usage: STAFInst [-source <source path>] [-target <target path>]
       [-bin <directory>] [-lib <directory>] [-preview]
       [-type < m | r | f >] [-verbose] [-warn]
       [-ro <mode>] [-rx <mode>] [-rw <mode>] [-dp <mode>]
       [-noreg | [-name <name>] [-email <email>] [-org <org>]]
       [-acceptlicense] [-option name=value]...
```

-source specifies the directory in which the STAF files were untar'd. In our example above, this would be /tmp/STAF/staf. The default is the current directory. If you are not running STAFInst from the directory in which STAFInst resides, you must specify -source.

-target specifies where STAF will be installed. The default is /usr/local/staf for all Unix systems except for Mac OS X where the default is /Library/staf.

-bin specifies the directory where softlinks to executables will be created. If this option is not specified, softlinks to executables will not be created.

-lib specifies the directory where softlinks to libraries will be created. If this option is not specified, softlinks to libraries will not be created.

-type specifies the type of installation (the default is rs). Possible values are:

- m - This installs the minimum files necessary to run STAF
- r - This installs all STAF files, including all supported codepages. This value is eqivalent to "f".
- f - This installs all STAF files, including all supported codepages. This value is eqivalent to "r".

-verbose specifies that all operations performed by STAFInst should be displayed to the user.

-warn specifies that STAFInst should stop execution if it encounters a warning condition. An example of a warning condition is trying to copy or link a non-existant file. Normally, these warning conditions are printed and execution continues.

-preview displays the intended actions of STAFInst without touching the file system.

-ro allows you to specify the mode bits for STAF files intended to be read-only. The default is 444.

-rx allows you to specify the mode bits for STAF files intended to be read-execute. The default is 555.

-rw allows you to specify the mode bits for STAF files intended to be read-write. The default is 664.

-dp allows you to specify the mode bits for the directories STAF creates. The default is 775.

-noreg prevents STAF from automatically registering with IBM Austin. This option is highly discouraged.

-name allows you to specify the name STAF uses when registering. The default is the empty string.

-email allows you to specify the e-mail address STAF uses when registering. The default is the empty string.

-org allows you to specify the organization STAF uses when registering. The default is the empty string.

-acceptlicense indicates that you accept the terms of the license agreement for this software. If not specified, the license agreement will be displayed at the beginning of the installation.

-option allows you to specify options for the installed components. The allowed name=value pairs are:

- **TCP=IPV4** Specifies to use the IPV4-only TCP libraries. This is the default.
- **TCP=IPV4_IPV6** Specifies to use the TCP libraries that support both IPv4 and IPv6.
- **USE_PERL_VERSION=5.8** Specifies to use Perl 5.8 as the default Perl support for STAF. This is the default. This option is only applicable for those platforms on which STAF supports Perl 5.8.
- **USE_PERL_VERSION=5.10** Specifies to use Perl 5.10 as the default Perl support for STAF. This option is only applicable for those platforms on which STAF supports Perl 5.10.
- **USE_PERL_VERSION=5.12** Specifies to use Perl 5.12 as the default Perl support for STAF. This option is only applicable for those platforms on which STAF supports Perl 5.12.
- **USE_PERL_VERSION=5.10** Specifies to use Perl 5.14 as the default Perl support for STAF. This option is only applicable for those platforms on which STAF supports Perl 5.14.
- **USE_PERL_VERSION=5.6** Specifies to use Perl 5.6 as the default Perl support for STAF. This option is only applicable for those platforms on which STAF supports Perl 5.6.
- **USE_PERL_SYSTEM_PATH=0** Specifies to not determine the version of Perl in the system PATH at install-time (if possible) and use that version of Perl by default. This is the default. Only valid on platforms with Perl support.
- **USE_PERL_SYSTEM_PATH=1** Specifies to determine the version of Perl in the system PATH at install-time (if possible) and use that

version of Perl by default. If there is no version of Perl in the system PATH (or if the output of "perl -v" returns an unexpected value, i.e. something other than "5.6.x", "5.8.x", "5.10.x", "5.12.x", or "5.14.x") then the default version of Perl (determined by the USE_PERL_VERSION option) will be used. Only valid on platforms with Perl support.

- **USE_PYTHON_VERSION=2.2** Specifies to use Python 2.2 as the default Python support for STAF. This is the default (except for Mac OS X and FreeBSD). This option is only applicable for those platforms on which STAF supports Python 2.2.
- **USE_PYTHON_VERSION=2.3** Specifies to use Python 2.3 as the default Python support for STAF. This is the default for Mac OS X. This option is only applicable for those platforms on which STAF supports Python 2.3.
- **USE_PYTHON_VERSION=2.4** Specifies to use Python 2.4 as the default Python support for STAF. This is the default for FreeBSD. This option is only applicable for those platforms on which STAF supports Python 2.4.
- **USE_PYTHON_VERSION=2.5** Specifies to use Python 2.5 as the default Python support for STAF. This option is only applicable for those platforms on which STAF supports Python 2.5.
- **USE_PYTHON_VERSION=2.6** Specifies to use Python 2.6 as the default Python support for STAF. This option is only applicable for those platforms on which STAF supports Python 2.6.
- **USE_PYTHON_VERSION=2.7** Specifies to use Python 2.7 as the default Python support for STAF. This option is only applicable for those platforms on which STAF supports Python 2.7.
- **USE_PYTHON_VERSION=3.0** Specifies to use Python 3.0 as the default Python support for STAF. This option is only applicable for those platforms on which STAF supports Python 3.0.
- **USE_PYTHON_VERSION=3.1** Specifies to use Python 3.1 as the default Python support for STAF. This option is only applicable for those platforms on which STAF supports Python 3.1.
- **USE_PYTHON_SYSTEM_PATH=0** Specifies to not determine the version of Python in the system PATH at install-time (if possible) and use that version of Python by default. This is the default. Only valid on platforms with Python support.
- **USE_PYTHON_SYSTEM_PATH=1** Specifies to determine the version of Python in the system PATH at install-time (if possible) and use that version of Python by default. If there is no version of Python in the system PATH (or if the output of "python -V" returns an unexpected value, i.e. something other than "2.2.x", "2.3.x", "2.4.x", "2.5.x", "2.6.x", "2.7.x", "3.0.x", or "3.1.x") then the default version of Python (determined by the USE_PYTHON_VERSION option) will be used. Only valid on platforms with Python support.
- **USE_TCL_VERSION=8.4** Specifies to use TCL 8.4 as the default TCL support for STAF. This option is only applicable for those platforms on which STAF supports TCL 8.4.
- **USE_TCL_VERSION=8.5** Specifies to use TCL 8.5 as the default TCL support for STAF. This option is only applicable for those platforms on which STAF supports TCL 8.5.
- **USE_TCL_VERSION=8.6** Specifies to use TCL 8.6 as the default TCL support for STAF. This option is only applicable for those platforms on which STAF supports TCL 8.6.
- **USE_TCL_SYSTEM_PATH=1** Specifies to determine the version of TCL in the system PATH at install-time (if possible) and use that version of TCL by default. If there is no version of TCL in the system PATH then the default version of TCL (determined by the USE_TCL_VERSION option) will be used. Only valid on platforms with TCL support.

IMPORTANT! In order to start STAF or to use STAF, you must set some environment variables as described in section 7 "Environment Variable Settings". STAFInst does not update these environment variables for you.

5.2. Examples using STAFInst

For the following examples, assume the STAF installation image is in /tmp/STAF and your current directory is also /tmp/STAF.

To take all the defaults, which will perform a recommended install of STAF into /usr/local/staf (or /Library/staf if installing on Mac OS X), you would simply enter

```
./STAFInst -acceptlicense
```

To perform a recommended install of STAF into the default target directory, and have softlinks placed into /usr/bin and /usr/lib, you would enter

```
./STAFInst -acceptlicense -bin /usr/bin -lib /usr/lib
```

To perform a minimum installation of STAF into /opt/staf, you would enter

```
./STAFInst -acceptlicense -target /opt/staf -type m
```

To perform a preview of a full STAF installation into the default target directory, you would enter

```
./STAFInst -acceptlicense -type f -preview
```

To perform a default install of STAF and provide your name and e-mail address for registration, you would enter

```
./STAFInst -acceptlicense -name "Your name" -email "Your e-mail address"
```

To install STAF in the default locations but change read-only permissions to 440, read-execute permissions to 550, read-write privileges to 660, and directory permissions to 770, you would enter

```
./STAFInst -acceptlicense -ro 440 -rw 550 -rw 660 -dp 770
```

To install STAF, create the -lib links, and use Perl 5.10, you would enter

```
./STAFInst -acceptlicense -lib /usr/lib -option USE_PERL_VERSION=5.10
```

To install STAF and specify the USE_PERL_SYSTEM_PATH option, you would enter

```
./STAFInst -acceptlicense -option USE_PERL_SYSTEM_PATH=1
```

To install STAF, create the -lib links, and specify the USE_PERL_SYSTEM_PATH and USE_PERL_VERSION options, you would enter

```
./STAFInst -acceptlicense -lib /usr/lib -option USE_PERL_SYSTEM_PATH=1 -option USE_PERL_VERSION=5.10
```

To install STAF and specify the USE_PYTHON_SYSTEM_PATH and USE_PYTHON_VERSION options, you would enter

```
./STAFInst -acceptlicense -option USE_PYTHON_SYSTEM_PATH=1 -option USE_PYTHON_VERSION=2.6
```

5.3. STAFInst Uninstall

When STAF is installed on Unix systems via STAFInst, a script called STAFUninst is created which can be used to uninstall STAF. This file is created in the directory where you installed STAF. All STAF features will be uninstalled; you cannot choose which features to uninstall. However, only the files and softlinks that the STAF install created will be removed. Any files created after STAF was installed will not be removed.

6. Platform Installation Notes

6.1. Linux installation

Note, when you are running STAF on Linux Red Hat 9.0 or later, you may encounter a Segmentation Fault every time you execute the STAF executable. To work around this problem, set the following environment variable:

```
LD_ASSUME_KERNEL=<kernel-version>
```

The following versions can be specified for <kernel-version>

- **2.4.1** - Linuxthreads with floating stacks
- **2.2.5** - Linuxthreads without floating stacks

For information on Linux operating system library compatibility, see section 10 "Operating System Library Compatability (Linux)".

When installing using IA on RHEL5 and other recent Linux distributions, when attempting to launch the STAF installer, the following error may be

seen: "The installer is unable to run in graphical mode." or "java.lang.UnsatisfiedLinkError: awt (libXp.so.6: cannot open shared object file: No such file or directory)". On RHEL5 and other recent Linux distributions, the libraries required to run a Swing based Java application (such as our STAF installers) may not be installed by default. See this InstallAnywhere article for more information. To resolve the problem, install the required X libraries on the Linux system. For example, for RHEL5, download libXp-1.0.0-8.1.el5.i386.rpm and install it via RPM.

Starting in STAF V3.4.1, the STAF InstallAnywhere bundled JVM installers for Linux use Java 6.0 (and in STAF V3.3.3 through STAF V3.4.0, the STAF InstallAnywhere bundled JVM installers for Linux used Java 5.0).

6.2. AIX installation

Starting in STAF V3.4.1, the STAF InstallAnywhere installer for AIX is bundled with a Java 6.0 JVM (and in STAF V3.3.4 through STAF V3.4.0, the STAF InstallAnywhere bundled JVM installers for AIX used Java 5.0). Java 5.0 and later is supported on AIX 5.2 or later.

Note that the current version of STAF for AIX requires AIX 6.1 or later.

6.3. HP-UX IA64 64-bit installation

Note, if you plan to use Java services, or STAF-enabled Java applications with the 64-bit version of STAF for HP-UX IA64, you must use Java 1.4.1 or later.

6.4. IBM i 32-bit (previously known as i5/OS or OS/400) installation

STAF V3.4.10 or later requires IBM i 7.1 or later (because STAF V3.4.10+ is built on AIX 6.1 and applications built on AIX 6.1 can only be run in PASE for i in IBM i 7.1 or later). Note that IBM i was previously known as i5/OS or OS/400. STAF for IBM i runs in PASE for i (Portable Application Solutions Environment for i), an AIX runtime environment on the IBM i operating system, so you will use the aix or aix64 STAF tar.gz installer file to install STAF on IBM i.

To install STAF on a IBM i system, perform the following steps:

1. Create a directory /QIBM/stafinst on your IBM i system (or use whatever directory name you prefer). This will be a temporary directory where you'll untar the STAF installer file.
2. Download latest version of STAF for IBM i (e.g. STAF34x-aix.tar.gz or STAF34x-aix64.tar.gz) from the Download STAF website to directory /QIBM/stafinst.
3. You must install STAF from the IBM i Portable Application Solutions Environment (IBM PASE for i). IBM PASE for i provides an AIX runtime environment on the IBM i operating system. Start an IBM PASE for i interactive terminal session by using the QP2TERM() program by typing the following from the IBM i command line.

   ```
   CALL QP2TERM
   ```

   The QP2TERM command writes the default Korn shell prompt (/QOpenSys/usr/bin/sh) to the screen.
4. Unzip the STAF34x-aix.tar.gz or STAF34x-aix64.tar.gz file and run the STAFInst executable to install STAF to location /QIBM/staf (or to

whatever location you prefer). See section 5 "Using STAFInst to install STAF" for more information on STAFInst. For example:

```
cd /QIBM/stafinst
gunzip STAF34x-aix.tar.gz
tar xf STAF34x-aix.tar
cd staf
. ./STAFInst -target /QIBM/staf -acceptlicense
```

Notes:

- If gunzip is not available on your IBM i machine and you have access to a AIX machine, you can use the AIX gunzip executable on the IBM i machine.
- If STAF is already installed in /QIBM/staf, and you want to upgrade to a new version of STAF, you should backup your STAF.cfg file and then run STAFUninst to uninstall STAF before running STAFInst, and then restore your backed up STAF.cfg file. For example, assuming you've already unzipped the STAF installer tar.gz file in /QIBM/stafinst, here are the steps to upgrade STAF:

```
cd /QIBM/staf
cp bin/STAF.cfg bin/STAF.cfg_save
. ./STAFUninst
cd /QIBM/stafinst/staf
. ./STAFInst -target /QIBM/staf -acceptlicense
cd /QIBM/staf
cp bin/STAF.cfg_save bin/STAF.cfg
```

5. You can now delete the temporary directory where you unzipped the STAF installer file, if desired.

Follow these instructions to start STAFProc:

1. Start an IBM i PASE interactive terminal session by using the QP2TERM() program by typing the following from the IBM i command line. Skip this step if you have already started an IBM i PASE interactive terminal session. You must run STAF in the IBM i PASE environment.

   ```
   CALL QP2TERM
   ```

   The QP2TERM command writes the default Korn shell prompt (/QOpenSys/usr/bin/sh) to the screen.
2. Change to the directory where you installed STAF and run STAFEnv.sh to set the necessary environment variables to use STAF:

   ```
   cd /QIBM/staf
   . ./STAFEnv.sh
   ```

   If you installed the 32-bit version of STAF, running STAFEnv.sh should set the following environment variables needed to use STAF:

```
export PATH=/QIBM/staf/bin:$PATH
export LIBPATH=/QIBM/staf/lib:$LIBPATH
export CLASSPATH=/QIBM/staf/lib/JSTAF.jar:{CLASSPATH}
export STAFCONVDIR=/QIBM/staf/codepage
export QIBM_MULTI_THREADED=Y
export QIBM_JAVA_PASE_STARTUP=/usr/lib/start32
```

If you installed the 64-bit version of STAF, the environment variables will be the same except for:

```
export QIBM_JAVA_PASE_STARTUP=/usr/lib/start64
```

3. Type "STAFProc &" to launch STAF in the background. Note probably you'll want to start STAFProc using nohup so that when you exit your connection to the IBM i machine, STAFProc will still be running. When using nohup, the output from STAFProc (including information on if an error occurred starting STAFProc) will be redirected to file nohup.out in the current directory (or you can specify to redirect the output to another file).

```
nohup STAFProc &
```

Note that the default shell for processes is /bin/sh. If you need to use the "qsh" shell for the IBM i PASE environment, then you will either need to specify

```
SHELL "/QOpenSys/usr/bin/qsh -c %C"
```

for PROCESS START requests, or change the default shell by adding the following line to your STAF.cfg file:

```
SET DEFAULTSHELL "/QOpenSys/usr/bin/qsh -c %C"
```

You must shutdown and restart STAFProc for any changes to the STAF.cfg file to take effect.

6.5. z/OS installation

Note: STAF will only run on z/OS V1.4 or newer. STAF's Java support on z/OS requires JDK 1.4.1 or newer.

To install STAF on z/OS, you should follow the steps described in section 5 "Using STAFInst to install STAF".

It is critical that you remember to make set the _CEE_RUNOPTS environment variable. Failure to set this variable will prevent STAF from working properly. The proper setting should be

```
_CEE_RUNOPTS="posix(on)"
```

If you did not install STAF in the default target directory (/usr/local/staf), STAFCONVDIR must be set to the STAF codepage directory.

6.6. FreeBSD installation

Support for the following programming languages is provided in the STAF installer files for FreeBSD:

- FreeBSD 7.4+ i386
  - Java 1.5+ support provided (compiled using JDK 1.5.0_16-p9)
  - Python support provided
  - No Perl or Tcl support

On FreeBSD platforms when using an InstallAnywhere installer file, if you get a Java OutOfMemoryError which prevents the installation from starting, you may need to increase the maximum Java heap size by setting the _JAVA_OPTIONS environment variable to 1 gigabyte before starting the installation as follows:

```
# export _JAVA_OPTIONS=-Xmx1g
# ./STAF3416-setup-freebsd.bin -i console
```

This Java OutOfMemoryError has been seen on FreeBSD when using Java 1.5. Another workaround when using the FreeBSD InstallAnywhere NoJVM installer file is to use Java 1.7 instead of Java 1.5 as the JVM.

6.7. Mac OS X installation

To use the .zip InstallAnywhere installer for Mac OS X, download the .zip file to the Mac OS X system, and double click on the .zip file. This will unzip the zip file and create a STAF333setup file in the directory containing the .zip file. To start the installation, double click on the STAF333setup file.

Note that silent and console installs for Mac OS X using the .zip InstallAnywhere installer are not currently supported. To perform a silent or console install, use the .bin InstallAnywhere installer.

6.8. Solaris installation

On Solaris Sparc 64-bit platforms if you get the following error when starting STAFProc:

```
/usr/sfw/lib/libstdc++.so.6: wrong ELF class: ELFCLASS32
```

The solution is to include "/usr/sfw/lib/sparcv9" in LD_LIBRARY_PATH:

```
LD_LIBRARY_PATH=/usr/sfw/lib/sparcv9:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
STAFProc &
```

/usr/sfw/lib/sparcv9 contains the 64-bit Solaris Sparc libraries.

Note that on Solaris, running "uname -a" on both x86 32-bit systems and AMD64/Opteron x64 systems will report "i86pc i386 i86pc". To determine whether you should use the Solaris x86 32-bit version of STAF or the Solaris AMD64/Opteron x64 version of STAF, you can run "isainfo -b". If the output is "32", then you need to use the Solaris x86 32-bit version of STAF. If the output is "64", then you need to use the Solaris AMD64/Opteron x64 version of STAF. Note that you can use the "psrinfo -pv" command to get more detailed information about the system architecture.

On Solaris x86 platforms when using an InstallAnywhere installer file, if you get a Java OutOfMemoryError which prevents the installation from starting, you may need to increase the maximum Java heap size by setting the _JAVA_OPTIONS environment variable to 1 gigabyte before starting the installation as follows:

```
# _JAVA_OPTIONS=-Xmx1g
# export _JAVA_OPTIONS
# ./STAF3416-setup-solaris-x86.bin -i console
```

This Java OutOfMemoryError has been seen on Solaris x86 when using Java 1.5. Another workaround when using the Solaris x86 InstallAnywhere NoJVM installer file is to use Java 1.7 instead of Java 1.5 as the JVM.

7. Environment Variable Settings

7.1. Environment Variable Settings

STAF requires setting some environment variables to run. If you performed an InstallAnywhere install and selected an option to have the install update the environment variables, these environment variables should already be updated for STAF. Otherwise, you must set these environment variables yourself. In most cases, you'll probably want to set these environments for the system so that they are always set even after logging out or rebooting. However, if you are going to be running two or more versions of STAF on the same machine, the install creates STAFEnv script files that you can use to easily switch environment settings for each version of STAF. See section 8, "Running Multiple Instances of STAFProc" for more information on using the STAFEnv script files when running multiple instances of STAF on the same machine.

7.2. Windows

- **PATH** must contain the STAF bin directory in order to start STAFProc or to communicate with STAF (e.g. C:\STAF\bin). Also, if you're going to be communicating with STAF from Java programs or registering STAF Java services, PATH should also contain the Java bin directory (e.g. C:\ibmjava60\bin).
- **CLASSPATH** must contain the JSTAF.jar file (e.g. C:\STAF\bin\JSTAF.jar). JSTAF.jar contains the STAF Java APIs to communicate with STAF from Java programs and is also required to register STAF services written in Java.
- **STAFCONVDIR** must be set to the STAF codepage directory if you did not install STAF in the default C:\STAF directory.
- **STAFCODEPAGE** must be set on systems which don't return a valid value from nl_langinfo(CODESET) (specifically if that call returns NULL or the empty string). The value of this variable should contain the codepage being used by the system. See codepage/alias.txt for valid values. If you are unsure, use "LATIN_1".

7.3. Unix

- **PATH** must contain the STAF bin directory in order to start STAFProc (e.g. /usr/local/staf/bin). Other windows need only have the STAF bin directory in the PATH if you did not allow softlinks to be created in the default system directories. Also, if you're going to be communicating with STAF from Java programs or registering STAF Java services, PATH should also contain the Java bin directory (e.g. /opt/IBMJava2-142/bin).
- **LD_LIBRARY_PATH** (**LIBPATH** on AIX, IBM i, and z/OS systems, **SHLIB_PATH** on HP-UX systems, **DYLD_LIBRARY_PATH** on Mac OS X) must contain the STAF lib directory (e.g. /usr/local/staf/lib), unless you allowed softlinks to be installed in the default system directories
- **CLASSPATH** must contain the JSTAF.jar file (e.g. /usr/local/staf/lib/JSTAF.jar). JSTAF.jar contains the STAF Java APIs to communicate with STAF from Java programs and is also required to register STAF services written in Java.
- **STAFCONVDIR** must be set to the STAF codepage directory if you did not install STAF in the default target directory (e.g. /usr/local/staf).
- **STAFCODEPAGE** must be set on systems which don't return a valid value from nl_langinfo(CODESET) (specifically if that call returns NULL or the empty string). This is known to be true on Solaris 2.6. The value of this variable should contain the codepage being used by the system. See codepage/alias.txt for valid values. If you are unsure, use "LATIN_1".
- (z/OS only) **_CEE_RUNOPTS** must be set to "posix(on)".

7.4. Mac OS X

- **PATH** must contain the STAF bin directory in order to start STAFProc (e.g. /Library/staf/bin).
- **DYLD_LIBRARY_PATH** must contain the STAF lib directory (e.g. /Library/staf/lib), unless you allowed softlinks to be installed in the default system directories
- **CLASSPATH** must contain the JSTAF.jar file (e.g. /Library/staf/lib/JSTAF.jar). JSTAF.jar contains the STAF Java APIs to communicate with STAF from Java programs and is also required to register STAF services written in Java.
- **STAFCONVDIR** must be set to the STAF codepage directory if you did not install STAF in the default target directory (e.g. /Library/staf).
- **STAFCODEPAGE** must be set on systems which don't return a valid value from nl_langinfo(CODESET) (specifically if that call returns NULL or the empty string). The value of this variable should contain the codepage being used by the system. See codepage/alias.txt for valid values. If you are unsure, use "LATIN_1".

7.5. IBM i (previously known as i5/OS or OS/400)

- **PATH** must contain the STAF bin directory in order to start STAFProc or to communicate with STAF (e.g. /QIBM/staf/bin). Also, if you're going to be communicating with STAF from Java programs or registering STAF Java services, PATH should also contain the Java bin directory.
- **LIBPATH** must contain the STAF lib directory (e.g. /QIBM/staf/lib).
- **CLASSPATH** must contain the JSTAF.jar file (e.g. /QIBM/staf/lib/JSTAF.jar). JSTAF.jar contains the STAF Java APIs to communicate with STAF from Java programs and is also required to register STAF services written in Java.
- **STAFCONVDIR** must be set to the STAF codepage directory (e.g. /QIBM/staf/codepage)
- **STAFCODEPAGE** must be set to LATIN_1.
- **QIBM_MULTI_THREADED** must be set to Y so that STAF can be accessed from the QSH (Qshell) shell. This environment variable is required since STAF is multi-threaded. If this environment variable is not set, the QSH STAF request will hang.
- **QIBM_JAVA_PASE_STARTUP** must be set to /usr/lib/start32 if you are using the 32-bit version of STAF, or set to /usr/lib/start64 if you are using the 64-bit version of STAF. This environment variable indicates to start a 32-bit or 64-bit IBM i PASE environment (needed when accessing the STAF Java support in QSH). If this environment variable is not set, an error such as the following will occur:

```
java.lang.ExceptionInInitializerError
    at java.lang.Throwable.<init>(Throwable.java:180)
    at java.lang.Error.<init>(Error.java:37)
    at java.lang.ExceptionInInitializerError.<init>(ExceptionInInitializerError.java:61)
    at echo.main(echo.java:10)
Caused by: java.lang.NullPointerException
    at java.lang.ClassLoader.loadLibrary0(ClassLoader.java:1640)
    at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1564)
    at java.lang.Runtime.loadLibrary0(Runtime.java:788)
    at java.lang.System.loadLibrary(System.java:834)
    at com.ibm.staf.STAFHandle.<clinit>(STAFHandle.java:100)
    ... 1 more
```

8. Installing Multiple Instances of STAF

8.1. Installing Multiple Instances of STAF

You can install different versions of STAF on the same machine in different locations. You can also run multiple instances of the same or different versions of STAF at the same time on a single machine.

See section "Running Multiple Instances of STAFProc" in the STAF User's Guide for more information on running multiple instances of STAF on a single machine.

9. Install/Uninstall Problems

9.1. Install Problems

If you have a problem while installing STAF, please check the STAF V3 FAQ, particularly section "STAF Install Questions".

If you have a problem starting STAFProc, please be sure that the necessary environment variables have been set for STAF (see section 7 "Environment Variable Settings"). Also, check the STAF V3 FAQ.

If you are getting an error such as "Error while loading shared libraries: libstdc++.so.6: cannot open shared object file: No such file or directory" when starting STAFProc on Linux, then see section 10 "Operating System Library Compatability (Linux)".

9.2. Uninstall Problems

There is a known intermittent problem where if you run the InstallAnywhere uninstaller, the uninstall will complete successfully and report no errors, but the STAF files have not been uninstalled (and on Windows the Start menu items for STAF have not been removed). If you are doing an upgrade install (which means that the uninstaller is executed prior to installing any new files), this does not cause the install to fail, since the files are overwritten. However, on Windows, you may end up with multiple Start menu entries. So if you upgrade from STAF 3.3.0 to STAF 3.4.16 on a system that has this problem, STAF 3.4.16 will be installed, and will work as expected. However, the STAF 3.3.0 entries in the Start menu ("Start STAF 3.3.0", "Shutdown STAF 3.3.0", "STAF 3.3.0") will still exist, in addition to the newly added "Start STAF 3.4.16", "Shutdown STAF 3.4.16", "STAF 3.4.16" entries. The old Start menu entries can just be deleted.

This is a known issue with InstallAnywhere due to a corrupt global registry. The problem can be resolved by renaming the InstallAnywhere registry file as documented here: "Q000085: INFO: Possible Causes for Common Uninstaller Problems"

After using the workaround, when you intsall STAF again, you should be able to successfully uninstall STAF. Note that on Windows you will need to manually delete any old Start menu entries for STAF.

10. Operating System Library Compatability (Linux)

10.1. Operating System Library Compatability (Linux)

Some platforms (such as Linux) on which we build and provide STAF are linked to C++/GCC operating system libraries. In previous versions of STAF, these operating system library files were packaged/installed with STAF. However, these library files could cause some incompatibility issues with the operating system and other native applications. Starting in STAF V3.2.0, these library files will no longer be packaged/installed with STAF. However, in order to run STAF on these platforms, you will need these library files or compatible library files.

As a STAF user, what this means is that you may need to make some one-time setup/configuration changes on your test machine(s) in order to get STAF V3.2.0 or later working correctly. There are several possible user scenarios:

- STAFProc starts successfully on your platform. This means that the operating system currently has compatible C++/GCC library support.
- When attempting to start "STAFProc" or "staf", you receive a message such as: "Error while loading shared libraries: libstdc++.so.6: cannot open shared object file: No such file or directory", "STAFProc: /usr/lib/libstdc++.so.6: no version information available (required by STAFProc)", or "STAFProc: relocation error: /usr/local/staf/lib/libSTAF.so: undefined symbol: _ZNSs4_Rep20_S_empty_rep_storageE" This means that the operating system does not have compatible C++/GCC libraries. There are several ways to resolve this problem:
  - Use the library files from our STAF build machines in the table below. It is up to the user to decide how to install the files on their operating system(s). In previous versions of STAF, these files were installed in /STAF/Config/STAFRoot}/lib (and the files were accessible due to this directory being included in the LD_LIBRARY_PATH environment variable).
  - Install the compatibility libraries for your operating system. The steps to install these libraries will vary depending upon the operating system and level. For example, on RHEL4, using the install media, under "Package Group Selection", select "Development" and then select "Legacy Software Development".

    - Build STAF on your operating system. Follow the instructions in the "STAF V3 Developer's Guide" (http://staf.sourceforge.net/current/stafdg.html).
- When attempting to start "STAFProc" or "staf", you receive a message such as: "/lib64/libgcc_s.so1: version `GCC_4.2.0` not found (required by /usr/local/staf/lib/libstdc++.so.6)". This means that the operating system has incompatible C++/GCC libraries in the library path. There are several ways to resolve this problem: e.g. Remove the /usr/local/staf/lib/... libraries if present.
- When using STAF for zlinux-32 or zlinux-64 on SLES11, if you are using IBM Java 1.4.2 for a STAF Java service or a Java application that uses the STAF Java APIs, the JVM may crash with an error similar to:

  ```
  JVMDG217: Dump Handler is Processing Signal 4 - Please Wait.
  JVMDG303: JVM Requesting Java core file
  JVMDG304: Java core file written to /javacore.20100115.163302.24679.txt
  JVMDG215: Dump Handler has Processed Exception Signal 4.
  ```

  To resolve the problem, install the libstdc++.so.6 files for zlinux-32 or zlinux-64 in the table below to the STAF lib directory (for example, /usr/local/staf/lib).

Here are the affected operating systems and details about the specific platforms on which we build STAF:

OS Name	STAF Installers	OS Details	Library file(s)
Linux IA32 (i386)	STAF34x-setup-linux.bin STAF34x-setup-linux-NoJVM.bin STAF34x-linux.tar.gz	Red Hat Enterprise Linux Server release 5.10 (Tikanga) gcc version 4.1.2 20080704 (Red Hat 4.1.2-54) Linux 2.6.18-371.3.1.el5 #1 SMP i686 i686 i386 GNU/Linux GNU C Library stable release version 2.5 Compiled by GNU CC version 4.1.2 20080704 (Red Hat 4.1.2-54). Compiled on a Linux 2.6.9 system on 2013-09-25.	libstdc++.so.6. libgcc_s.so.1.
Linux AMD64 (x86_64)	STAF3416-setup-linux-amd64.bin STAF3416-setup-linux-amd64-NoJVM.bin STAF3416-linux-amd64.tar.gz	Red Hat Enterprise Linux Server release 5.10 (Tikanga) gcc version 4.1.2 20080704 (Red Hat 4.1.2-54) GNU Make 3.81 Linux 2.6.18-371.3.1.el5 #1 SMP x86_64 x86_64 x86_64 GNU/Linux GNU C Library stable release version 2.5. Compiled by GNU CC version 4.1.2 20080704 (Red Hat 4.1.2-54). Compiled on a Linux 2.6.9 system on 2013-09-25.	libstdc++.so.6.

Linux PPC64 (32-bit Java support)	STAF34x-setup-linux-ppc64-32.bin STAF34x-setup-linux-ppc64-32-NoJVM.bin STAF34x-linux-ppc64-32.tar.gz	SUSE LINUX Enterprise Server 10 (ppc) VERSION=10 PATCHLEVEL=4 gcc version 4.1.2 20070115 (SUSE Linux) Linux 2.6.16-60-0.99.1-ppc64 ppc64 ppc64 ppc64 GNU/Linux GNU C Library stable release version 2.4 (20121101) Configured for ppc-suse-linux. Compiled by GNU CC version 4.1.2 20070115 (SUSE Linux). Compiled on a Linux 2.6.16 system on 2012-11-01.	libstdc++.so.5. libgcc_s.so.1.
Linux PPC64 (64-bit Java support)	STAF34x-setup-linux-ppc64-64.bin STAF34x-setup-linux-ppc64-64-NoJVM.bin STAF34x-linux-ppc64-64.tar.gz	SUSE LINUX Enterprise Server 10 (ppc) VERSION=10 PATCHLEVEL=4 gcc version 4.1.2 20070115 (SUSE Linux) Linux 2.6.16-60-0.99.1-ppc64 ppc64 ppc64 ppc64 GNU/Linux GNU C Library stable release version 2.4 (20121101) Configured for ppc-suse-linux. Compiled by GNU CC version 4.1.2 20070115 (SUSE Linux). Compiled on a Linux 2.6.16 system on 2012-11-01.	libstdc++.so.5. libgcc_s.so.1.
zLinux 31-bit (zlinux-32)	STAF34x-setup-zlinux-32.bin STAF34x-setup-zlinux-32-NoJVM.bin STAF34x-zlinux-32.tar.gz	SUSE Linux Enterprise Server 10 (s390x) VERSION = 10 gcc version 4.1.0 (SUSE Linux) Linux 2.6.16.21-0.8-default s390x s390x s390x GNU/Linux GNU C Library development release version 2.4 (20060616) Configured for s390-suse-linux. Compiled by GNU CC version 4.1.0 (SUSE Linux). Compiled on a Linux 2.6.16 system on 2006-06-16.	libstdc++.so.6.
zLinux 64-bit (zlinux-64)	STAF34x-setup-zlinux-64.bin STAF34x-setup-zlinux-64-NoJVM.bin STAF34x-zlinux-64.tar.gz	SUSE Linux Enterprise Server 10 (s390x) VERSION = 10 gcc version 4.1.0 (SUSE Linux) Linux 2.6.16.21-0.8-default s390x s390x s390x GNU/Linux GNU C Library development release version 2.4 (20060616) Configured for s390-suse-linux. Compiled by GNU CC version 4.1.0 (SUSE Linux). Compiled on a Linux 2.6.16 system on 2006-06-16.	libstdc++.so.6.

11. Starting STAFProc during system reboot

11.1. Unix

You can have STAFProc start automatically during reboot on Unix machines by using a script file similar to the following:

```
#!/bin/sh
PATH=/usr/local/staf/bin:$PATH
export PATH
LD_LIBRARY_PATH=/usr/local/staf/lib
export LD_LIBRARY_PATH
CLASSPATH=/usr/local/staf/lib/JSTAF.jar:/usr/local/staf/samples/demo/STAFDemo.jar
export CLASSPATH
STAFCONVDIR=/usr/local/staf/codepage
export STAFCONVDIR
STAFCODEPAGE=LATIN_1
export STAFCODEPAGE
nohup /usr/local/staf/bin/STAFProc > /usr/local/staf/stafproc.out &
```

As an alternative, you can use a script file that simply executes the startSTAFProc.sh script, which will set the appropriate environment variables and start STAFProc via nohup:

```
#!/bin/sh
/usr/local/staf/startSTAFProc.sh &
```

See the STAFEnv.sh and startSTAFProc.sh files provided in the STAF root directory (e.g. /usr/local/staf). The STAFEnv.sh file contains the correct environment variables settings for STAF for your machine.

If you did not install STAF to the default directory (/usr/local/staf), then you would need to specify the non-default installation directory in the script file.

Note that on AIX, you need to replace LD_LIBRARY_PATH with LIBPATH, and on HP-UX, you need to replace LD_LIBRARY_PATH with SHLIB_PATH.

Note that if your Unix operating system does not support the "nohup" command, then remove it from the last line in the script:

```
/usr/local/staf/bin/STAFProc > /usr/local/staf/stafproc.out &
```

The console output from STAFProc will be stored in the /usr/local/staf/stafproc.out file.

The usage of this script file varies depending on the specific Unix operating system:

- **Linux**

  On Linux, there are multiple ways that you can start applications during the system boot process. In general, we recommend that you create a file called /etc/rc.staf with contents:

  ```
  #!/bin/sh
  /usr/local/staf/startSTAFProc.sh &
  ```

  Make the file executeable (755 is most likely sufficient)

  ```
  chmod 777 /etc/rc.staf
  ```

  Next add the following line to /etc/inittab:

  ```
  staf:5:boot:/etc/rc.staf
  ```

  Other methods of starting STAFProc may cause system programs located in /sbin (such as "fdisk", "ifconfig", etc.) to work incorrectly when executed via STAF, which is why we recommend using /etc/rc.staf and /etc/inittab as documented above.

  Other ways to start STAFProc during boot on Linux include adding the script lines to the /etc/rc.d/rc.local file, or on Linux SuSe, adding the script lines to the /etc/init.d/boot.local file. You should select the method that works best for your environment.

  Note that we recommend starting STAFProc under the root user id. To start STAFProc under the root user id, change the last line above to the following:

  ```
  su -c '/usr/local/staf/bin/STAFProc' - root &
  ```

  If starting STAFProc under the root user id, you may also need to modify /root/.bash_profile by adding /usr/local/staf/bin to the PATH environment variable as follows:

  ```
  PATH=$PATH:$HOME/bin:/usr/local/staf/bin
  ```

Here is some more information on other ways to start STAFProc during boot-up on some specific Linux distributions:

- **RHEL 4 and 5**

  On Red Hat Enterprise Linux (RHEL) 4 and 5, you can create an init script for stafproc that can be used to start STAFProc automatically when booting up a Linux RHEL system and to shutdown STAFProc when shutting down a Linux RHEL system.

  For more information on init scripts, see Managing Initscripts with Red Hat's chkconfig, or google for additional documentation.

  1. Create an init script that can be used in conjunction with the launch script /etc/rc.d/rc. Create a file named stafproc with the following contents and put it in directory /etc/rc.d/init.d. You can customize it as needed.

```
#!/bin/bash
#
# stafproc:    Starts the STAFProc Daemon
#
# chkconfig: 2345 90 30
# description: Starts/stops STAFProc
# processname: stafproc

STAF_HOME=/usr/local/staf
STAF_INSTANCE_NAME=STAF

# Include any directories you want in STAFProc's PATH
PATH=/opt/IBMJava2-142/bin:/usr/local/sbin:/usr/local/bin:/root/bin:$PATH
export PATH

. $STAF_HOME/STAFEnv.sh $STAF_INSTANCE_NAME

start() {
        echo "Starting STAFProc: "

        # Check if STAFProc is already running
        STAF local PING PING > /dev/null
        if [ $? -eq 0 ]; then
          echo "STAFProc is already running with STAF_INSTANCE_NAME=
$STAF_INSTANCE_NAME"
          return 0
        fi
```

```
        STAFProc >> $STAF_HOME/STAFProc.out &
        RETVAL=$?

        if [ $RETVAL -eq 0 ]; then
          echo "Waiting for STAFProc to finish initalizing..."
          # Ping every 2s (up to 30 attempts) until returns RC 0
          count=1
          while [ $count -le 30 ]; do
            sleep 2
            STAF local PING PING > /dev/null
            RETVAL=$?
            if [ $RETVAL -eq 0 ]; then
              touch /var/lock/subsys/stafproc
              echo "STAFProc initialized"
              count=30
            fi
            ((count ++))
          done
        fi
        echo
        return $RETVAL
}

stop() {
        echo "Shutting down STAFProc: "

        # Kill any STAF requests that are currently running to speed up
        # the shutdown of STAF and to ensure that the submitters are
        # notified that the STAF requests were terminated.
        #
        # Note: Could be using the uppercase STAF or lowercase staf
        #       executable so submit a killall command for each.
        killall -9 STAF > /dev/null 2>&1
        killall -9 staf > /dev/null 2>&1

        # Shutdown STAF (stops any STAF services and STAFProc
        STAF local SHUTDOWN SHUTDOWN > /dev/null
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
```

```
            rm -f /var/lock/subsys/stafproc

            echo "Waiting for STAFProc to finish shutdown..."
            # Ping every 2s (up to 60 attempts) until get RC 21
            # which means "STAF not running"
            count=1
            while [ $count -le 60 ]; do
              sleep 2
              STAF local PING PING > /dev/null
              RETVAL=$?
              if [ $RETVAL -eq 21 ]; then
                echo "STAFProc ended normally"
                RETVAL=0
                count=60
              fi
              ((count ++))
            done
          elif [ $RETVAL -eq 21 ]; then
            echo "STAFProc is not currently running so shutdown is ignored"
          fi

          echo
          return $RETVAL
}

restart() {
          stop
          start
}

RETVAL=0

# See how we were called.
case "$1" in
  start)
          start
          ;;
  stop)
          stop
          ;;
```

```
  restart|reload)
        restart
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart}"
        exit 1
esac

exit $?
```

Note that this init script assumes that STAF was installed in the default location of /usr/local/staf. You should customize the PATH environment variable setting in the init script for your environment. This example added a Java bin directory to the PATH so that STAFProc can run Java executables and Java services without fully qualifying the path to the java executable. Also, note that this init script redirects STAFProc's output to file /usr/local/staf/STAFProc.out. You can customize that as needed.

Note that this init script contains the following line:

```
# chkconfig: 2345 90 30
```

This tells the **chkconfig --add** how to behave in relation to this script.

- The first number, 2345, tells it that the service should be active for run states 2,3,4 and 5 (that it will run the stafproc init script passing it the **start** argument. For the remaining run states, the service should be off (e.g. will run the stafproc init script passing it the **stop** argument.
- The second number, 90, tells it that the Start priority should be 90.
- The last number, 30, tells it that the Kill priority should be 30.

Note that the init script creates a file named stafproc in /var/lock/subsys to show that the service is up and running when the service is started and it deletes this file when the service is stopped. This is needed for the script to be run correctly.

2. Make the init script executable by running the following (as root):

   ```
   # chmod +x /etc/rc.d/init.d/stafproc
   ```

3. Verify that the init script works correctly for your environment by running it manually. Try it passing it the "stop", "start", and "restart" options

   ```
   # cd /etc/rc.d/init.d
   ```

```
# ./stafproc stop
Shutting down STAFProc:
Waiting for STAFProc to finish shutdown...
STAFProc ended normally

# ./stafproc start
Starting STAFProc:
Waiting for STAFProc to finish initalizing...
STAFProc initialized

# ./stafproc restart
Shutting down STAFProc:
Waiting for STAFProc to finish shutdown...
STAFProc ended normally

Starting STAFProc:
Waiting for STAFProc to finish initalizing...
STAFProc initialized
```

4. Add the init script to the chkconfig configuration as follows:

   ```
   # chkconfig --add stafproc
   ```

   Verify that the stafproc init script has been added:

   ```
   # chkconfig --list | grep stafproc
   stafproc        0:off   1:off   2:on    3:on    4:on    5:on    6:off
   # find /etc/rc.d -name '*stafproc' -print
   /etc/rc.d/rc0.d/K30stafproc
   /etc/rc.d/rc2.d/S90stafproc
   /etc/rc.d/rc3.d/S90stafproc
   /etc/rc.d/rc6.d/K30stafproc
   /etc/rc.d/rc4.d/S90stafproc
   /etc/rc.d/init.d/stafproc
   /etc/rc.d/rc1.d/K30stafproc
   /etc/rc.d/rc5.d/S90stafproc
   ```

5. Now verify that STAFProc starts up automatically when the system boots up and shuts down when the system is shutdown. Note that when the stafproc init script is run during boot-up and system shutdown, you can view the output from the stafproc

init script in file /var/log/messages.

- **RHEL 6**

  On Red Hat Enterprise Linux (RHEL) 6, the method used to start programs during system boot has changed. Instead of updating inittab, you use upstart. Here is some documentation on using upstart.

  Create a file called /etc/init/startstaf.conf (the file name does not matter, but the extension must be ".conf") with the following content:

  ```
  expect daemon
  exec /usr/local/staf/startSTAFProc.sh
  start on stopped rcS
  ```

  Of course, if you didn't install STAF in /usr/local/staf, then you would modify the second line. Also, you could use "start on runlevel 5" for the third line. Then after rebooting, STAFProc should be already started.

- **SLES 10 and 11**

  For SUSE Linux Enterprise Server (SLES) 10 and 11, follow these instructions:

  1. Create your own /etc/init.d/rc.local file as root.

  2. Add the following lines to the rc.local file (substituting the actual location where you installed STAF, if you did not use the default /usr/local/staf):

     ```
     #!/bin/sh
     ### BEGIN INIT INFO
     # Provides: rc.local
     # Required-Start: $network $syslog
     # Required-Stop: $network $syslog
     # Default-Start: 3 5
     # Default-Stop: 0 1 2 6
     # Description: whatever
     ### END INIT INFO
     /usr/local/staf/startSTAFProc.sh &
     ```

  3. Run the following commands:

     ```
     chmod 755 rc.local
     ```

```
chkconfig --add rc.local
```

- **Fedora 15 and later**

  For Fedora 15 and later, follow these instructions to enable STAFProc to start during system boot:

  1. Create a file named startstaf.service in directory /lib/systemd/system with the following contents (substituting the actual location where you installed STAF, if you did not use the default /usr/local/staf):

     ```
     [Unit]
     Description=Start the STAF service
     After=syslog.target

     [Service]
     ExecStart=/usr/local/staf/startSTAFProc.sh
     Type=forking

     [Install]
     WantedBy=multi-user.target graphical.target
     ```

  2. As root execute the following command:

     ```
     systemctl daemon-reload
     ```

  3. Check if the service definition was loaded by executing:

     ```
     systemctl status startstaf.service
     ```

     If the service definition loaded OK you should see some output similar to this:

     ```
     STAF.service - 'STAF Process'
          Loaded: loaded (/etc/systemd/system/startstaf.service)
          Active: inactive (dead)
          CGroup: name=systemd:/usr/local/staf/startSTAFProc.sh
     ```

  4. If the service definition was loaded successfully, execute the following commands:

     ```
     systemctl start startstaf.service
     systemctl enable startstaf.service
     ```

The STAFProc process should now be running and will be launched automatically at system boot from now on.

5. To view the status of the running service at any time execute the following command:

   ```
   systemctl status startstaf.service
   ```

   If the process has launched and is running, this command will output a message similar to this:

   ```
   startstaf.service - Start the STAF service
             Loaded: loaded (/lib/systemd/system/startstaf.service)
             Active: active (running) since Mon, 23 Jan 2012 14:36:27 +0000; 10min ago
            Process: 608 ExecStart=/usr/local/staf/startSTAFProc.sh (code=exited,
   status=0/SUCCESS)
           Main PID: 621 (STAFProc)
             CGroup: name=systemd:/system/startstaf.service
                     621 /usr/local/staf/bin/STAFProc
   ```

- **FreeBSD**

  On FreeBSD, you can create an init script for stafproc that can be used to start STAFProc automatically when booting up a FreeBSD system and to shutdown STAFProc when shutting down a FreeBSD system.

  For more information on FreeBSD init scripts, google for additional documentation.

  1. Create an init script that can be used in conjunction with the launch script /etc/rc.d/rc. Create a file named stafproc with the following contents and put it in directory /etc/rc.d. You can customize it as needed.

     ```
     #!/bin/sh
     #
     # PROVIDE: stafproc
     # REQUIRE: networking filesystems
     # KEYWORD: shutdown
     #
     # Add the following line to /etc/rc.conf to enable STAFProc
     #
     # stafproc_enable="YES"
     #
     ```

```
. /etc/rc.subr

name="stafproc"
start_cmd="stafproc_start"
stop_cmd="stafproc_stop"
stafproc_enable="yes"
rcvars=stafproc_enable

STAF_HOME=/usr/local/staf
STAF_INSTANCE_NAME=STAF

# Include any directories you want in STAFProc's PATH
PATH=/usr/local/jdk1.5.0/bin:$PATH
export PATH

stafproc_start()
{
    echo "Starting STAFProc..."
    . $STAF_HOME/STAFEnv.sh $STAF_INSTANCE_NAME

    # Check if STAFProc is already running
    STAF local PING PING > /dev/null
    if [ $? -eq 0 ]; then
        echo "STAFProc is already running with STAF_INSTANCE_NAME=$STAF_INSTANCE_NAME"
        return 0
    fi

    STAFProc >> $STAF_HOME/STAFProc.out &
    RETVAL=$?

    echo "STAFProc output will be written to /usr/local/staf/STAFProc.out"

    if [ $RETVAL -eq 0 ]; then
        echo "Waiting for STAFProc to finish initalizing..."
        # Ping every 2s (up to 30 attempts) until returns RC 0
        count=1
        while [ $count -le 30 ]; do
            sleep 2
            STAF local PING PING > /dev/null
            RETVAL=$?
```

```
            if [ $RETVAL -eq 0 ]; then
                echo "STAFProc initialized"
                count=30
            fi
            count=$(($count+1))
        done
    fi

    echo
    return $RETVAL
}

stafproc_stop()
{
    echo "Shutting down STAFProc..."

    # Shutdown STAF (stops any STAF services and STAFProc)
    . $STAF_HOME/STAFEnv.sh $STAF_INSTANCE_NAME
    STAF local SHUTDOWN SHUTDOWN
    RETVAL=$?
    if [ $RETVAL -eq 0 ]; then
        echo "Waiting for STAFProc to finish shutdown..."
        # Ping every 2s (up to 60 attempts) until get RC 21
        # which means "STAF not running"
        count=1
        while [ $count -le 60 ]; do
            sleep 2
            STAF local PING PING > /dev/null
            RETVAL=$?
            if [ $RETVAL -eq 21 ]; then
                echo "STAFProc ended normally"
                RETVAL=0
                count=60
            fi
            count=$(($count+1))
        done
    elif [ $RETVAL -eq 21 ]; then
        echo "STAFProc is not currently running so shutdown is ignored"
    fi
```

```
    echo
    return $RETVAL
}

load_rc_config $name
run_rc_command "$1"
```

Init Script Customization Notes:

a. Change the STAF_HOME variable setting if you installed STAF to a location other than the default location of /usr/local/staf,
b. This sample init script redirects STAFProc's output to file $STAF_HOME/STAFProc.out (e.g. /usr/local/staf/STAFProc.out). You can change the file name where you want STAFProc's output redirected if needed.
c. Customize the PATH environment variable if needed. This sample init script adds a Java bin directory to the PATH environment variable as follows so that if STAFProc registers any Java services it can find the java executable without specifying the JVM option. Customize the PATH to include the location of your JVM (or any other directories you want to have in STAFProc's PATH) or remove these lines if you don't need to change the PATH environment variable.

```
# Include any directories you want in STAFProc's PATH
PATH=/usr/local/jdk1.5.0/bin:$PATH
export PATH
```

2. Make the init script executable by running the following (as root):

```
# chmod +x /etc/rc.d/stafproc
```

3. Verify that the stafproc init script works correctly for your environment by running it manually. Try it passing it the "stop", "start", and "restart" options as follows:

```
# /etc/rc.d/stafproc stop
Shutting down STAFProc...
Response
--------

Waiting for STAFProc to finish shutdown...
STAFProc ended normally

# /etc/rc.d/stafproc start
Starting STAFProc...
STAFProc output will be written to /usr/local/staf/STAFProc.out
Waiting for STAFProc to finish initalizing...
```

```
STAFProc initialized

# /etc/rc.d/stafproc restart
Shutting down STAFProc...
Response
--------

Waiting for STAFProc to finish shutdown...
STAFProc ended normally

Starting STAFProc...
STAFProc output will be written to /usr/local/staf/STAFProc.out
Waiting for STAFProc to finish initalizing...
STAFProc initialized

#
```

4. Enable the stafproc service by editing file /etc/rc.conf and add the following line:

   ```
   stafproc_enable="YES"
   ```

5. Now verify that STAFProc starts up automatically when the system boots up and shuts down when the system is shutdown. Check the STAFProc output file (e.g. /usr/local/staf/STAFProc.out) to see messages logged when STAFProc starts up or is shut down normally.

- **Solaris**

  On Solaris, you can start STAFProc at boot-up using the init.d startup mechanism or you can use the Service Management Facility (SMF) added in Solaris 10. Here's a description of both of these startup methods:

  - To use the init.d startup mechanism to start STAFProc at boot-up, follow these steps:

    Run the following command, and keep track of the path returned:

    ```
    which sh
    ```

    For example, the path might be /usr/bin/sh.

    Create a file in the /etc/init.d directory called something like "startSTAF". Note that the file should not end in .sh or anything else. Edit your file and add the script lines to it (using the "sh" path from the previous step as the first line in the script):

```
#!/usr/bin/sh

PATH=/usr/local/staf/bin:/jdk1.3.1/bin:${PATH:-}; export PATH
CLASSPATH=/usr/local/staf/lib/JSTAF.jar:${CLASSPATH:-}; export CLASSPATH
STAFCONVDIR=/usr/local/staf/codepage; export STAFCONVDIR
LD_LIBRARY_PATH=/usr/local/staf/lib:${LD_LIBRARY_PATH:-}; export LD_LIBRARY_PATH
/usr/local/staf/bin/STAFProc &
```

Make the file executable (755 is most likely sufficient)

```
chmod 777 /etc/init.d/startSTAF
```

Symlink the file to the rc3.d directory and start STAFProc (being sure to use nohup):

```
ln -s /etc/init.d/startSTAF /etc/rc3.d/S98startSTAF
nohup /etc/rc3.d/S98startSTAF start
```

Notes:

- The prefix S means default state Started for Solaris
- 98 is just some number
- The suffix is the filename from init.d

- To use SMF (Service Management Facility) to start STAFProc at boot-up, follow these steps. For more information about SMF, see the links provided by the OpenSolaris Community Group and the Using Solaris SMF article.

1. Use your favorite editor to create a file named stopSTAFProc.sh in the directory where STAF is installed (e.g. /usr/local/staf) with the following contents:

   ```
   #!/bin/sh
   # Shuts down STAFProc
   . /usr/local/staf/STAFEnv.sh
   STAF local SHUTDOWN SHUTDOWN
   ```

   If STAF is not installed in the default location of /usr/local/staf, you'll need to edit it to change /usr/local/staf to the directory where STAF is installed.

   Make the script file executable (755 is most likely sufficient):

```
# chmod 777 /usr/local/staf/stopSTAFProc.sh
```

The SMF service manifest you're going to create next will use the startSTAFProc.sh script provided by STAF to start STAFProc at boot-up and will use this stopSTAFProc.sh script to shutdown STAFProc when the system is shutdown.

2. Make a directory named /var/svc/manifest/application/staf and use your favorite editor to create a SMF service manifest XML file named stafproc.xml in that directory with the following contents.

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="stafproc">
  <service name="application/staf/stafproc" type="service" version="1">
    <dependency name="network" grouping="require_all" restart_on="error"
type="service">
      <service_fmri value="svc:/milestone/network:default"/>
    </dependency>
    <dependency name="filesystem" grouping="require_all" restart_on="none"
type="service">
      <service_fmri value="svc:/system/filesystem/local"/>
    </dependency>
    <instance name="default" enabled="false">
      <method_context>
        <method_credential user="root" group="daemon"/>
        <method_environment>
          <envvar name="PATH" value="/usr/sfw/bin:/usr/sbin:/usr/bin"/>
        </method_environment>
      </method_context>
      <exec_method type="method" name="start" exec="/usr/local/staf/startSTAFProc.sh"
timeout_seconds="0"/>
      <exec_method type="method" name="stop" exec="/usr/local/staf/stopSTAFProc.sh"
timeout_seconds="60"/>
      <property_group name="startd" type="framework">
        <propval name="duration" type="astring" value="wait"/>
      </property_group>
      <property_group name="application" type="application"/>
    </instance>
    <stability value="External"/>
    <template>
      <common_name>
```

```
        <loctext xml:lang="C">STAFProc</loctext>
      </common_name>
      <documentation>
        <doc_link name="documentation" uri="http://staf.sourceforge.net"/>
      </documentation>
    </template>
  </service>
</service_bundle>
```

If STAF is not installed in the default location of /usr/local/staf, you'll need to edit it to change all instances of /usr/local/staf to the directory where STAF is installed. Also, you may need to customize the PATH environment variable value.

This is just a sample manifest for starting/stopping STAFProc. You can customize it as needed.

3. Validate the manifest xml file and import it into the SMF system:

```
# cd /var/svc/manifest/application/staf
# svccfg validate stafproc.xml
# svccfg import stafproc.xml
```

Note that if an error occurs and you need to start fresh with a new stafproc.xml file, you can return back to a clean state as follows:

```
# svccfg delete stafproc
```

4. Enable the stafproc SMF service:

```
# svcadm enable stafproc
```

5. Check the log file to verify that STAFProc was started successfully:

```
# tail /var/svc/log/application-staf-stafproc:default.log
[ Jun 26 16:36:28 Enabled. ]
[ Jun 26 16:36:28 Executing start method ("/usr/local/staf/startSTAFProc.sh") ]
[ Jun 26 16:36:28 Ignoring duplicate environment variable "PATH=/usr/sbin:/usr/bin". ]
[ Jun 26 16:36:28 Method "start" exited with status 0 ]

Machine          : staf3b.austin.ibm.com
Machine nickname : staf3b.austin.ibm.com
```

```
Startup time     : 20130626-16:36:28

STAFProc version 3.4.14 initialized
```

6. You can use the Solaris **svcs** command to check the that the service is now enabled and online:

```
# svcs -l stafproc
fmri         svc:/application/staf/stafproc:default
name         STAFProc
enabled      true
state        online
next_state   none
state_time   Wed June 26 16:36:28 2012
logfile      /var/svc/log/application-staf-stafproc:default.log
restarter    svc:/system/svc/restarter:default
contract_id  114
dependency   require_all/error svc:/milestone/network:default (online)
dependency   require_all/none svc:/system/filesystem/local (online)
# svcs -x stafproc
svc:/application/staf/stafproc:default (STAFProc)
 State: online since Wed Jun 26 16:36:28 2012
   See: /var/svc/log/application-staf-stafproc:default.log
Impact: None.
```

7. Verify that STAFProc is up and running by submitting a local STAF PING request:

```
# STAF local PING PING
Response
--------
PONG
```

8. Disable the service to shutdown STAFProc and verify that is working properly:

```
# svcadm disable stafproc
```

Check the log file to verify that STAFProc was shut down properly:

```
# tail /var/svc/log/application-staf-stafproc:default.log
STAFProc version 3.4.14 initialized
```

```
20130626-16:38:42;17;00000100;STAFProcess::processMonitorThread: Error opening /dev/
tty, errno: 6
[ Jun 26 16:39:21 Stopping because service disabled. ]
[ Jun 26 16:39:21 Executing stop method ("/usr/local/staf/stopSTAFProc.sh") ]
[ Jun 26 16:39:21 Ignoring duplicate environment variable "PATH=/usr/sbin:/usr/bin". ]
STAFProc ending normally
Response
--------

[ Jun 26 16:39:21 Method "stop" exited with status 0 ]
```

9. Enable the stafproc SMF service again so that STAFProc is left running:

   ```
   # svcadm enable stafproc
   ```

Now STAFProc will be automatically started when the system boots up and shutdown when the system is shutdown.

- **AIX**

  On AIX, create a /etc/rc.xxxx file (such as /etc/rc.staf), which contains the script lines to set the STAF environment variables and to start STAFProc. For example:

  ```
  #!/bin/sh
  PATH=/usr/local/staf/bin:$PATH
  export PATH
  LIBPATH=/usr/local/staf/lib
  export LIBPATH
  CLASSPATH=/usr/local/staf/lib/JSTAF.jar:/usr/local/staf/samples/demo/STAFDemo.jar
  export CLASSPATH
  STAFCONVDIR=/usr/local/staf/codepage
  export STAFCONVDIR
  STAFCODEPAGE=LATIN_1
  export STAFCODEPAGE
  /usr/local/staf/bin/STAFProc > /usr/local/staf/stafproc.out &
  ```

  Then add an entry to the /etc/inittab file so that the rc.staf file will execute during the boot. For example, add the following line to the end of /etc/inittab:

  ```
  staf:2:boot:/etc/rc.staf
  ```

See the AIX documentation for more information on updating the /etc/inittab file.

Note that if you are using STAF Java services, then you must make sure the file system where the JVM is located is available during the AIX system boot. For example, if STAF is installed to /usr/local/staf, but the JVM being used for the STAF Java services is in the /opt file system, then the /opt file system may not be mounted when the rc.staf file is executed, and the stafproc.out file may contain an error such as:

```
27:Error constructing service, JSTAF, Result: Error starting a process for the JVM using
JVMName: STAFJVM1, JVM: /opt/aix64java/jre/bin/java, JVMOptions:  -Xmx5
12m , RC: 10, Result: Invalid command: /opt/aix64java/jre/bin/java
The command is not a file or does not have execute permissions. OS RC 2  Make sure /opt/
aix64java/jre/bin/java exists.
```

To prevent this error, make sure that the JVM resides in a file system (such as /usr) that will be available during system boot.

- **HP-UX**

On HP-UX, follow these steps to start STAFProc automatically during the boot process:

1. Create a file named something like "staf" in /sbin/init.d and add the script lines to set STAF environment variables and to start STAFProc that were specified at the beginning of this section to this file.

   Note: You may also need to add /bin to the PATH environment if it isn't already in the PATH so that the nohup command can be found.

2. Create a file named "staf" (use the same name as the file you created in /sbin/init.d) in /etc/rc.config.d and add a variable declaration like:

   ```
   # STAF: Set to 1 to start STAFProc daemon
   #
   STAF=1
   ```

   Give this file execute permissions so that this file can be executed on startup.

   ```
   chmod 755 /etc/rc.config.d/staf
   ```

3. Create a symbolic link to your /sbin/init.d/staf file called S996staf in /sbin/rc3.d. Note that the suffix (e.g. "staf") in your file name must match what you named your file in /sbin/init.d and the prefix of "S996" in the file name allows the CDE server to start before STAFProc.

```
ln -s /sbin/init.d/staf /sbin/rc3.d/S996staf
```

You may get the following error message in the output when STAFProc is started:

```
STAFProcess::processMonitorThread: error opening /dev/tty, errno: 6
```

However, STAFProc does start despite this message, and functions normally.

11.2. Mac OS X

Note that there are multiple ways to configure a Mac OS X system to start an application during the boot process. This sections shows an example of one way to start STAFProc during reboot on Mac OS X (this example assumes that the steps are done via the root/admin userid).

In directory /Library/LaunchDaemons create a file called com.ibm.staf.plist.

In the com.ibm.staf.plist file add:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
    "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
        <string>com.ibm.staf</string>
    <key>Program</key>
        <string>/Library/staf/startSTAFProc.sh</string>
    <key>RunAtLoad</key>
    <true/>
    <key>StandardOutPath</key>
        <string>/Library/staf/com.ibm.staf.out</string>
    <key>StandardErrorPath</key>
        <string>/Library/staf/com.ibm.staf.out</string>
</dict>
</plist>
```

Of, course, change /Library/staf to the actual location where you installed STAF, if you didn't use the default install location.

When the system is rebooted, STAFProc should be started, and the STAFProc output will be in file /Library/staf/com.ibm.staf.out.

Note that if the com.ibm.staf.out file shows the "Machine" and "Machine nickname" as "localhost" instead of the actual hostname (and STAF

variables STAF/Config/Machine and STAF/Config/MachineNickname are set to "localhost"), then the network services most likely had not completed configuring when STAF was started during the reboot. To resolve this issue, add "ipconfig waitall" to the startSTAFProc.sh script, prior to where it starts STAFProc. Here is the updated startSTAFProc.sh:

```
#!/bin/sh
# Sets up the STAF environment variables and starts STAFProc
# in the background, logging STAFProc output to nohup.out
ipconfig waitall
. /Library/staf/STAFEnv.sh
nohup /Library/staf/bin/STAFProc &
```

Note that if you make this change, you will probably want to create you own copy of startSTAFProc.sh, and update the "Program" key to reference the copy of the script (since the /Library/staf/startSTAFProc.sh file will be overwritten if you reinstall/upgrade STAF).

Note that on Mac OS X Snow Leopard (10.6), you will need to make some additional updates.

- In your copy of startSTAFProc.sh, change line "nohup /Library/staf/bin/STAFProc &" to "/Library/staf/bin/STAFProc" (remove "nohup" and "&").
- You may need to add "sleep 60" (or some number of seconds) after the "ipconfig waitall" to make sure that STAFProc is using the correct hostnames.
- Here is an example of the updated startSTAFProc.sh file for Mac OS X Snow Leopard:

```
#!/bin/sh
# Sets up the STAF environment variables and starts STAFProc
# in the background, logging STAFProc output to nohup.out
ipconfig waitall
sleep 60
. /Library/staf/STAFEnv.sh
/Library/staf/bin/STAFProc
```

11.3. Windows

If you wish to have STAFProc start automatically on Windows when the operating system is rebooted (without requiring a user to log on to Windows), you can install STAF as a Windows service.

These instructions assume you have copied INSTSRV.EXE and SVRANY.EXE (available from the Windows Resource Kit) into C:\WINNT\SYSTEM32.

1. Create a base STAF service which by default is started automatically when the system boots.

   ```
   instsrv STAF c:\winnt\system32\srvany.exe
   ```

2. Start the Windows Registry Editor by typing regedit and find:

   ```
   My Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\STAF
   ```

3. Create a new string value with the name Description. Set the value to Automatically start STAF service.

4. Create a new Key with name Parameters. Inside the new key, create a new string value with the name Application and set the value to c:\staf\bin\stafproc.exe.

5. Close the Registry Editor.

6. If on Windows 2000, open the following program:

   ```
   Start > Programs > Administrative Tools > Services
   ```

7. If on Windows 2000 Terminal Server, open the following program:

   ```
   Start > Programs > Administrative Tools > Management Console (Windows 2000 Terminal Server)
   Services and Applications > Services
   ```

8. If on Windows XP Professional, open the following program:

   ```
   Start > Settings > Control Panel > Administrative Tools > Services
   ```

9. Locate STAF, right click on the STAF entry, and select Properties.

10. From the 'Log On' tab, click 'Allow service to interact with desktop', then click on Apply and then click on OK.

11. Next, type:

    ```
    net start staf
    ```

12. Check that STAFProc is now running from a command prompt:

    ```
    staf local ping ping
    ```

13. Note that when running STAF as a Windows service, you should not have STAFProc automatically start when users log on to Windows. You should either deselect the "Start STAF on user login" option when installing STAF, or delete STAF from Start > Programs > Startup folder.

14. Note that STAF should not be started or shutdown with the commands on the start menu when STAF is configured as a service.

15. Note that if you have STAF Java services configured in your STAF.cfg file when STAFProc is started as a Windows service and you log off the user id that you were logged on as when starting the STAF Windows service, you may find that STAF Java service requests will then fail with RC 6 (because the java.exe used to run the STAF Java services was terminated). If so, then you will need to pass the -Xrs parameter to the JVM used for your Java services (to tell the JVM to ignore the logoff signal/event). You can do this by updating your STAF.cfg file to add "OPTION J2=-Xrs" when registering a STAF Java service that creates a new JVM. For example:

    ```
    SERVICE STAX LIBRARY JSTAF EXECUTE {STAF/Config/STAFRoot}/services/stax/STAX.jar \
                 OPTION JVMName=STAX OPTION J2=-Xrs
    ```

    Then you can restart the STAF Windows service and this problem should no longer occur if you logoff.

As an alternative to configuring STAFProc as a Windows service, you can start STAFProc as a scheduled task. This works for Windows 2000 or later. This is especially useful for Windows 2003 64-bit machines which do not support srvany.exe in the Windows Toolkit (used when starting STAF as a Windows service). Note that when starting STAFProc as a scheduled task, it does not display any dialog. The following instructions are for starting STAFProc as a scheduled task on Windows 2003.

1. From the Start menu, select Control Panel > Scheduled Tasks > Add Scheduled Tasks.

2. In the "Scheduled Task Wizard", click on "Next" on the first panel.

3. In the list of applications, scroll to "Start STAF 3.3.5" (or whatever version you have installed) and select it. Note that "Start STAF x.x.x" should appear on the list of applications if you performed a recommended install of STAF using the defaults. If it is not in the list, click on "Browse" to select your STAFProc executable. Click on "Next".

4. On the next panel give the task a name, select "When my computer starts", and click on "Next".

5. For "Enter the user name:", enter "NT Authority\System" (without the double quotes, and remember that there is a space between "NT" and "Authority" and leave the password fields blank (otherwise you have to reset the password every time the controlling user resets their password). No password is required in this panel, but you need to be an administrator to set it up. Click on "Next".

6. Select "Open advanced properties for this task when I click Finish" and then click on "Finish".

7. In the "Settings" tab, de-select "Stop the task if it runs for...". Click on "Apply" and then "OK".

8. When you install STAF on a system where you plan to run it as a Scheduled Task, you should de-select the option that says "Start STAF when a user logs in". Otherwise, when a user logs in, it will try to start STAFProc again which causes problems. If you have already installed STAF, remove it from Start > Programs > Startup for all users to prevent STAFProc from starting again when a user logs in.

9. Reboot the machine and STAFProc should automatically start (without any command dialogs or anyone having to log in). You can view the tasks that you have scheduled by selecting Control Panel > Scheduled Tasks. In the submenu you should see the names of the scheduled tasks.