BEA Jolt

Installation and Configuration Guide

Jolt 1.2 Release
Document Edition 1.2
October 1999

Copyright

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, ObjectBroker, TOP END, and TUXEDO are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA Manager, BEA MessageQ, Jolt, M3, and WebLogic Enterprise are trademarks of BEA Systems, Inc.

All other company names may be trademarks of the respective companies with which they are associated.

BEA Jolt Developer’s Guide

Document Edition	Part Number	Date	Software Version
1.2	801-001104-001	October 1999	BEA Jolt 1.2

Contents

Preface

1. Preparing to Install Jolt

2. Installing Jolt

3. Configuring the Jolt System

4. Post Installation

Index

Preface

Purpose of This Manual

This manual describes how to install and configure the BEA Jolt ™ product. (Hereafter, unless the version is specified, “Jolt” refers to Jolt 1.2.)

Audience

This document is intended for system administrators, network administrators, and developers installing Jolt. The Jolt documentation presumes you are familiar with BEA TUXEDO and Java programming.

How This Manual is Organized

This manual is organized as follows:

Chapter 1, “Preinstallation,” discusses things you need to consider before you begin the installation procedure.

Chapter 2, “Installing Jolt,” describes the installation procedure for all applicable operating systems.

Chapter 3, “Configuring Jolt,” explains the configuration process.

Chapter 4, "Post Installation," instructs you in things to consider after installing and configuring Jolt.

Note: BEA TUXEDO and BEA Jolt are trademarked terms. Any occurrence of the terms TUXEDO and Jolt in the document, refers to the BEA TUXEDO and BEA Jolt products.

Jolt Documentation

The Jolt documentation includes:

BEA Jolt Installation and Configuration Guide (this document) (available in hardcopy format and online in HTML and PDF formats)

BEA Jolt Developer's Guide (available online in HTML and PDF formats)

BEA Jolt Release Notes (available in hardcopy format)

Jolt API Reference in Javadoc

Related Manuals

TUXEDO System Reference Manual

TUXEDO System Administration Guide

TUXEDO System Programmer's Guide, Volumes 1 and 2

TUXEDO System Message Manual, Volumes 1 and 2

Using Jolt for WebLogic

BEA Documentation Website

You can now access BEA documentation online at: e-docs.beasys.com.

Other TUXEDO Resources

Andrade, Juan, Mark Carges, Terence Dywer, and Stephen Felts. *The TUXEDO System: Software for Constructing and Managing Distributed Business Applications.* Reading, Mass.: Addison-Wesley, 1996.

Hall, Carl. *Building Client/Server Applications Using TUXEDO*. New York: John Wiley & Sons, 1996.

Primatesta, Fulvio. *TUXEDO: An Open Approach to OLTP*. Englewood Cliffs, N. J.:Prentice Hall, 1995.

Document Conventions

The following documentation conventions are used throughout this manual:

Item	Convention	Example
Arguments	Appear in parentheses and are formatted in a lowercase, monospace font. Optional arguments are formatted in italic font. Predefined arguments are formatted in an uppercase font.	(`*name*, 0, *value*`) (`ACCTID, 2, 5000`)

Caution	Applies to practices that could result in the loss of information.	**Caution:** Be sure to save your information before moving to the next window.
Environment variables	Are formatted in an uppercase font.	ENVFILE=${APPDIR}
Key names	Are presented in boldface type.	Press **Enter** to continue.
Literals	Are formatted in a monospace font.	`class extendSample`
Notes	Highlight procedures and contain information that assist the user in understanding the information contained in this manual.	**Note:** This feature is available with Jolt.
Programs and applications	Are formatted with initial caps.	Use the Administration Console.
User input	Is formatted in a monospace font.	Type `cd TUXDIR`
Warning	Applies to practices that could result in the loss of productivity or information.	**Warning:** Be sure to save your information before returning to the previous screen.
Window items	Are presented in boldface type. Window items can be window titles, button labels, text edit box names or other parts of the window.	Type your password in the **Logon window**. Select **Export** to make the service available to the client.
Code samples and command-line options may use the following conventions:		
`[ ]`	Square brackets contain arguments that are optional.	`[-d]`

`<>`	In ASCII files angle brackets contain a variable that must be replaced by an appropriate value.	`<filename>`

1 Preparing to Install Jolt

This chapter provides information on the preparations you need to make prior to installing BEA Jolt, hereafter designated as Jolt. The following topics are discussed in this chapter:

- What's New in Jolt
- What's Changed in Jolt
- Checking Your Package
- System Requirements
- Release Migration/Interoperability
- Jolt ASP Connectivity for TUXEDO
- Directory Structure
- Preinstallation Checklist

What’s New in Jolt

The following features are new to Jolt:

- **Compression**—Allows application data that is sent between a Jolt client and a Jolt Server (JSH) to be compressed during transmission over the network at a very low cost (few CPU cycles). Compression is only supported for clients based on JDK 1.1.7 or 1.2.
- **Diffie-Hellman (DH) Key Exchange**—Replaces the DES key exchange mechanism. This allows Jolt to be classified as a true 128 bit secure environment (available to U. S. customers only).
- **Two Session Keys per Jolt Session**—Jolt uses two session keys for each user session. One key is used for encrypting and decrypting messages sent from the client to the server, the other for messages sent from the server to the client.
- **Support for WebLogic Enterprise 4.2 (WLE)**—WLE 4.2 provides advanced Java development services. It allows you to build distributed, mission-critical, CORBA-compliant applications in Java and C++.
- **JRLY as an NT Service**—The Jolt Relay (JRLY) in Jolt on the NT platform is provided as an NT service. As an NT service, this process can be started and stopped via the NT Service Control Manager (SCM).
- **JRLY Connection to JRAD**—The Jolt Relay (JRLY) in Jolt allows you to specify a list of IP addresses for JRADs. On startup, the JRLY tries to connect to each JRAD on the list, searching sequentially from the beginning. The first JRAD to respond successfully is used.
- **Multiple JSL Addresses for JoltSessionAttributes**—Jolt allows you to specify a list of IP port addresses from which the Jolt client randomly selects JSLs until a connection is established.
- **ASP Connectivity for TUXEDO** (formerly Jolt WAS for IIS)—Works with existing Microsoft Internet Information Server (IIS) to provide a gateway for HTML clients into a TUXEDO application environment. Interactions between the Web server and Jolt classes are through VBScript and VB inside ASPs.
- **JSE Connectivity for TUXEDO** (formerly Jolt WAS for Servlets)—Simplifies handling of HTTP requests to the TUXEDO application environment.

- **Internationalization**—Allows all informational and error messages that are generated by the Jolt client to be localized.
- **Javadoc Format**—The HTML and PDF versions of the Jolt class library are now in Javadoc format.
- **Y2K Compliance**—Any dates that are used or displayed by Jolt are in Y2K-compliant format. Specifically, the Jolt license file allows expiration dates beyond December 31, 1999 and all messages and timestamps that Jolt generates use a Y2K-compliant representation.
- **Linux**—Jolt now supports the Linux operating system.
- **OS 400 and OS 390**—Jolt now supports these operating systems.

What's Changed in Jolt

If you are migrating from Jolt 1.1 to Jolt 1.2, you should be aware of the following changes:

- **TUXEDO/WLE Version**—If you are currently running TUXEDO 6.1, 6.2, or 6.3 and are upgrading to Jolt 1.2, you must also upgrade your TUXEDO version to 6.4 or 6.5, or WLE 4.2. Jolt 1.2 only works with versions 6.4 and 6.5 of TUXEDO, or version 4.2 of WLE.
- **Client/Server Interoperability**—Jolt provides some client/server interoperability. While you must upgrade all server-side components together, you do not have to upgrade client components at the same time. A Jolt 1.1 client and Jolt 1.2 server interoperate, but the client has only 1.1 functionality; server components have additional functionality. However, a Jolt 1.2 client cannot operate with a Jolt 1.1 server.
- **ASP Connectivity for TUXEDO**—The name of Jolt Web Application Server (WAS) has changed to ASP Connectivity for TUXEDO to emphasize the Microsoft ASP environment. Also, ASP Connectivity for TUXEDO is no longer an add-on; it is part of the Jolt Client Class Library.
- **JSE Connectivity for TUXEDO**—The name of Jolt WAS for Servlets is now JSE Connectivity for TUXEDO.

- **JoltBeans—**This feature is no longer an add-on; it is part of the Jolt Client Class Library.
- **Security/Encryption—**Diffie-Hellman (DH) key exchange replaces the DES key exchange mechanism (for U. S. customers only).
- **Digital Alpha NT—**Is no longer supported.

Checking Your Package

You should find the following contents when you open the Jolt box:

- Jolt CD (With either U.S. or International encryption)
- Jolt Documentation CD—Includes:
    - HTML and PDF versions of Jolt *Installation and Configuration Guide*
    - HTML and PDF versions of Jolt *Developer's Guide*
    - *Jolt API Reference in Javadoc*
- *Jolt Release Notes*—A hardcopy document that informs you of changes in the software that occurred after release.
- *Jolt Installation and Configuration Guide* (this document)—A hardcopy document (also in PDF on the Documentation CD) that provides instructions for installing and configuring *Jolt*.

The license is available in two formats:

- A 3.5-inch diskette that contains the product license may be taped outside the box
- A license file can be emailed to you directly.

System Requirements

The following hardware and software components are required before you can install Jolt.

BEA TUXEDO

If you are using Jolt with BEA TUXEDO, you must have TUXEDO 6.4 or 6.5 installed before you can install Jolt. TUXEDO versions prior to 6.4 are not supported by Jolt 1.2.

WebLogic Enterprise

If you plan to use Jolt with WebLogic Enterprise (WLE), you must have WLE 4.2 installed before you install Jolt.

Web Servers Supported

To provide HTTP/HTML-based access to enterprise applications, Jolt supports the Microsoft Internet Information Server (IIS) or Java Web Server.

Supported Platforms for Jolt Server

Jolt server platform support depends upon TUXEDO version support, as stated earlier in this chapter. The Jolt server also requires:

- CD-ROM access
- 500K of disk space

Table 1-1 shows the supported platforms for the Jolt server.

Table 1-1 Supported Platforms for Jolt Server

Operating System Vendor	Operating System	Operating System Version	TUXEDO Version Supported	WLE Version Supported
Digital Equipment Corporation (DEC)	Digital UNIX	4.0d	6.5 only	
DEC	Digital UNIX	4.0e		4.2
Hewlett-Packard (HP)	HP-UX	10.20	6.4 and 6.5	4.2
HP	HP-UX	11.0	6.4 and 6.5	4.2
International Business Machines (IBM)	AIX	4.2.1 (RS/6000)	6.4 and 6.5	
IBM	AIX	4.3.1 (RS/6000)	6.4 and 6.5	
IBM	AIX	4.3.2		4.2
IBM	OS/390	V2R6	6.5 only	
IBM	OS/400	4.1 (RISC)	6.5 only	
Microsoft	Windows NT	4.0 (sp4)	6.4 and 6.5	4.2
Red Hat	Linux	5.2	6.5 only	
Santa Cruz Operation (SCO)	Unixware	7.0	6.5 only	
Sequent	Dynix/PTS	4.4.2	6.5 only	4.2
SGI (Silicon Graphics)	IRIX	6.4 -o32	6.4 and 6.5	
SGI	IRIX	6.5 -n32	6.5 only	4.2
Sun	Solaris	2.5.1 (SPARC)	6.4 and 6.5	
Sun	Solaris	2.5.1 (Intel)	6.4 and 6.5	

Table 1-1 Supported Platforms for Jolt Server (Continued)

Operating System Vendor	Operating System	Operating System Version	TUXEDO Version Supported	WLE Version Supported
Sun	Solaris	2.6 (SPARC)	6.4 and 6.5	4.2
Sun	Solaris	2.6 (Intel)	6.4 and 6.5	
Sun	Solaris	2.7 (SPARC)	6.5 only	4.2
Sun	Solaris	2.7 (Intel)	6.5 only	
Siemens Nixdorf (SNI)	Reliant UNIX	5.4.4b	6.5 only	

Supported Platforms for Jolt Relay

Table 1-2 shows the supported platforms for Jolt Relay:

Table 1-2 Supported Platforms for Jolt Relay

Operating System Vendor	Operating System	Operating System Version
DEC	Digital UNIX	4.0d (Alpha)
HP	HP-UX	10.20
HP	HP-UX	11.00
IBM	AIX	4.2.1 (RS/6000)
IBM	AIX	4.3, 4.3.1
Microsoft	Windows NT	4.0 (Intel) (sp4)
Sequent	Dynix/PTS	4.4.2
SNI	Reliant UNIX	5.4.4b
Sun	Solaris	2.5.1 (SPARC)
Sun	Solaris	2.6 (SPARC)

Client Support

Jolt 1.2 supports Java applets and standalone Java client applications for TUXEDO. The following table shows the client types supported by Jolt 1.2.

Table 1-3 Client Types Supported by Jolt 1.2

Client Type	Supports	Version Supported
Java Applet in a Browser	Microsoft Internet Explorer	Internet Explorer (IE) 4.0, 5.0
	Netscape	Netscape Communicator 4.5.1
Standalone Applications on a Desktop	Java	JDK 1.1.7, 1.2

The HTML-based Jolt Client Classes run inside a Web server. The following Web servers are certified with Jolt.

Table 1-4 Web Servers Certified With Jolt

Vendor	Web Server Version	OS Version	Jolt Client Personality	Comments
Microsoft	IIS 4.0x	Windows NT 4.0	ASP Connectivity for TUXEDO	Supports VB or VBScript-based applications inside ASPs.
BEA WebXpress	WebLogic (WLX and/or WLS) 4.xx	Any	WebLogic Connectivity for TUXEDO	Supports servlets. Backward-compatible only - You can have a Jolt 1.2 server with a Jolt 1.1 client.
Any	Java Servlet Engine (Any version)	Any OS running JDK 1.1.7	JSE Connectivity for TUXEDO	Supports servlets.

Jolt Client Requirements

Jolt has the following client requirements:

- 574K of disk space for client classes.
- 1364K of disk space for client API documentation.
- 190K of disk space for client examples.
- Java Developer's Kit (JDK) 1.1.7 or 1.2 is certified for Jolt 1.2 application development. Jolt 1.1 uses JDK 1.1.5 or 1.1.6. (http://java.sun.com:80/java.sun.com/products/JDK/index.html)
- Java-enabled browser (Internet Explorer) or Java Virtual Machine (JVM).

Jolt Client Class Library

Various implementations of Java tend to show minor differences in characteristics. Jolt 1.2 is based on the 1.1.7 JDK. The Jolt 1.2 class library is compatible with JDK versions 1.1.7 and 1.2 on the following operating systems:

- Solaris 2.5.1, 2.6, and 2.7
- Windows NT 4.0
- Windows 95 and 98
- OSR2

The Jolt class library is compatible with the browsers and JDK versions shown in Table 1-5.

Table 1-5 Jolt Class Library Compatibility

Vendor	Browser/Version	Java Virtual Machine (JVM)	OS Version
Microsoft	Internet Explorer 4.0.1 (sp1)	MS JVM 4.0	Windows NT 4.0 Service Pack 4

Table 1-5 Jolt Class Library Compatibility (Continued)

Vendor	Browser/Version	Java Virtual Machine (JVM)	OS Version
Microsoft	Internet Explorer 5.0	MS JVM 4.0	Windows NT 4.0 Service Pack 4
Netscape	Communicator 4.5, 4.6.1	JDK 1.1.5	Windows NT 4.0

Release Migration/Interoperability

If you have Jolt 1.1 installed, we recommend that you uninstall it before you install Jolt 1.2.

Jolt provides increased interoperability between versions. In the Jolt 1.2 release, however:

- You must upgrade all server-side components together.
- If you have Jolt 1.1 applications, only Jolt 1.1 functionality is available to the Jolt 1.1 client, even though the server-side components have additional functionality when they are upgraded to Jolt 1.2.
- A Jolt 1.2 client cannot operate with a Jolt 1.1 server.

Jolt ASP Connectivity for TUXEDO

Requirements

Before you install Jolt ASP Connectivity for TUXEDO, you must have the following software installed:

- Windows NT Server 4.0 with NT Option Pack
- Microsoft IIS 4.0
- TUXEDO 6.x

Installation Instructions

Note: These instructions are documented from the Microsoft standard. Check the Microsoft documentation for updates.

To install Jolt ASP Connectivity for TUXEDO:

1. Remove any files that you copied when you installed the Jolt 1.1 WAS package.
2. Install the Microsoft Java Component Framework from the IIS 4.0 SDK (if it is not already installed).
3. Create a new directory, `aspcomp` in the `java\Trustlib` directory. (This is most typically found in `%windir%\java\TrustLib`).

   Since the framework files have been created in a package called `aspcomp`, the Java Virtual Machine (JVM) expects to find them there.

   The Microsoft Java Component Framework files are typically found in: `C:\InetPub\iissamples\sdk\components\java\Framework`
4. Copy the framework class files to the newly created

   `java\TrustLib\aspcomp` directory.

   (You only need to copy the *.class files.)

5. Install the latest Jolt 1.2 classes in the Windows NT server `java\Trustlib` directory.

   Contact BEA Customer Support for the latest Jolt 1.2 patch files.

6. Copy the class subdirectories and files (or unzip jolt.zip) from the Jolt 1.1 Patch 5 client distribution into your `java\lib` directory. The Jolt 1.2 classes have been created in packages beginning with “bea” so the JVM expects to find them in the `java\lib\bea\...` subdirectories.

   Install the corresponding Jolt 1.2 patch files on your Tuxedo server if you have not already done so.

7. Install the Web Application Services classes in the Windows NT server

   `java\Trustlib` directory.

8. Unzip `webiis.zip` to create the class files in your `java\trustlib` directory.

   Make sure you expand the files using the subdirectory folders in the zip archive.

   The Web Application Services classes are created in packages beginning with “bea” so the JVM will expect to find them in the `java\Trustlib\bea\...` subdirectories.

9. Register the Web Application Service classes as ActiveX components.

   Run the `wasreg.cmd` command file to register the BEA Web Application Services java classes as BEAWEB ActiveX components. This enables the BEAWEB components to be accessible from the Microsoft ASP scripts.

References

Refer to the online Microsoft NT Option Pack Product Documentation, especially the Microsoft Internet Information Server (IIS) chapters.

Directory Structure

The following sections show the CD directory structures for UNIX and NT systems.

UNIX Directory Structure on Jolt CD

Table 1-6 shows you directory paths for components of specified platforms on the Jolt CD.

Table 1-6 Examples of Directory Paths for UNIX Platforms

Platform	Directory Path
UNIX installation program	`/unix/ install.sh`
Alpha	
Jolt with TUXEDO 6.5 on Alpha dux40	`/unix/alpha/dux40/TUX6.5/jolt`
Jolt with WLE 4.2 on Alpha dux40	`/unix/alpha/dux40/WLE4.2/jolt`
HP	
Jolt with TUXEDO 6.4 or 6.5 on HP hpux1020	`/unix/hp/hpux1020/TUX6.4_6.5 /jolt`
Jolt with TUXEDO 6.4 or 6.5 on HP hpux11	`/unix/hp/hpux11/TUX6.4_6.5/jolt`
Jolt with WLE 4.2 on HP hpux1020	`/unix/hp/hpux1020/WLE4.2/jolt`
Jolt with WLE 4.2 on HP hpux11	`/unix/hp/hpux11/WLE4.2/jolt`
IBM	
Jolt with TUXEDO 6.4 or 6.5 on IBM aix42	`/unix/ibm/aix42/TUX6.4_6.5/jolt`

Table 1-6 Examples of Directory Paths for UNIX Platforms (Continued)

Platform	Directory Path
Jolt with TUXEDO 6.4 or 6.5 on IBM aix43	`/unix/ibm/aix43/TUX6.4_6.5/jolt`
Jolt with WLE 4.2 on IBM aix42	`/unix/ibm/aix43/WLE4.2/jolt`
Linux	
Jolt with TUXEDO 6.4 or 6.5 on Linux inlnx52	`/unix/linux/inlnx52/TUX6.4_6.5 /jolt`
Sequent	
Jolt with TUXEDO 6.5 on Sequent dynix 442	`/unix/seq/dynix442/TUX6.5/jolt`
Jolt with WLE 4.2 on Sequent dynix 442	`/unix/seq/dynix442/WLE4.2/jolt`
SGI	
Jolt with TUXEDO 6.4 or 6.5 on SGI irix64	`/unix/sgi/irix64/TUX6.4_6.5/jolt`
Jolt with TUXEDO 6.5 on SGI irix65	`/unix/sgi/irix65/TUX6.5/jolt`
Jolt with WLE 4.2 on SGI irix65	`/unix/sgi/irix65/WLE4.2/jolt`
SNI	
Jolt with TUXEDO 6.5 on SNI rel544	`/unix/sni/rel544/TUX6.5/jolt`
Sun	
Jolt with TUXEDO 6.4 or 6.5 on Sun SPARC Solaris 2.5.1	`/unix/ sun5x/sol251/TUX6.4_6.5/jolt`
Jolt with TUXEDO 6.4 or 6.5 on Sun SPARC Solaris 2.6	`/unix/ sun5x/sol26/TUX6.4_6.5/jolt`
Jolt with WLE 4.2 on Sun SPARC Solaris 2.6	`/unix/ sun5x/sol26/WLE4.2/jolt`

Table 1-6 Examples of Directory Paths for UNIX Platforms (Continued)

Platform	Directory Path
Jolt with TUXEDO 6.5 on Sun SPARC Solaris 2.7	`/unix/ sun5x/sol27/TUX6.5/jolt`
Jolt with WLE 4.2 on Sun SPARC Solaris 2.7	`/unix/ sun5x/sol27/WLE4.2/jolt`
Jolt with TUXEDO 6.4 or 6.5 on Sun Intel Solaris 2.5.1	`/unix/ sun5x/isol251/TUX6.4_6.5/jolt`
Jolt with TUXEDO 6.4 or 6.5 on Sun Intel Solaris 2.6	`/unix/ sun5x/isol26/TUX6.4_6.5/jolt`
Jolt with TUXEDO 6.5 on Sun Intel Solaris 2.7	`/unix/ sun5x/isol27/TUX6.5/jolt`
UW	
Jolt with TUXEDO 6.5 on UnixWare uw70	`/unix/uw/uw70/TUX6.5/jolt`

NT Directory Structure on Jolt CD

Table 1-7 shows the Jolt CD directory structure for NT platforms. On NT Intel, the installation is automatically launched when you insert the CD.

Table 1-7 Examples of Directory Path for NT Platform

Platform	Directory Path
Jolt on Windows NT 4.0 (Intel)	`jolt1.2/inwnt40/setup.exe`

IBM Mainframe Directory Structure on Jolt CD

Table 1-8 shows the Jolt CD directory structure for IBM OS 400 and OS 390 platforms.

Table 1-8 Examples of Directory Paths for IBM Mainframes

Platform	Directory Path
Jolt on OS400	`jolt1.2/as400/us/AS400/V4R1/README.txt`
Jolt on OS390	`jolt1.2/os390/os390.tar`

Preinstallation Checklist

Before you install Jolt, check the following:

- Back up your existing `jrepository` and `jrly.config` files.
- Uninstall previous version of Jolt, if applicable. Note that you can install all Jolt 1.2 components at once or separately, in the same or different directories. However, a single uninstall removes everything: files, directories and registry entries.
- Verify that you have TUXEDO version 6.4 or 6.5 or WLE 4.2 installed, if applicable.
- Determine the location of the TUXEDO directory where the Jolt server is to be installed.
- Determine the location of the directory where the Jolt documentation is to be installed.
- Determine the Web server location where the Jolt client components are to be installed.
- Verify the user ID and group ID assigned to Jolt server files (on UNIX).
- Verify the user ID and group ID to be assigned to Jolt client files (on UNIX).
- Review the *BEA Jolt Release Notes* and the Jolt Product Page (http://www.beasys.com) for any new information .

Caution: Jolt 1.2 automatically installs two Microsoft dynamic link libraries (DLLs), `MSVCRT.DLL` and `MFC42U.DLL` and overwrites older versions of these libraries. Before you begin installation, check whether older versions of these dynamic link libraries already exist. If they do exist and you do not want them to be overwritten, back them up.

CHAPTER

2 Installing Jolt

This chapter explains how to install Jolt software and online documentation.

This chapter includes the following sections:

- Microsoft Windows NT Installation Instructions
- UNIX/Linux System Installation Instructions
- OS 390 Installation Instructions
- OS 400 Installation Instructions
- Licensing Jolt for TUXEDO 6.4 and 6.5
- Installing Jolt Online Documentation

Microsoft Windows NT Installation Instructions

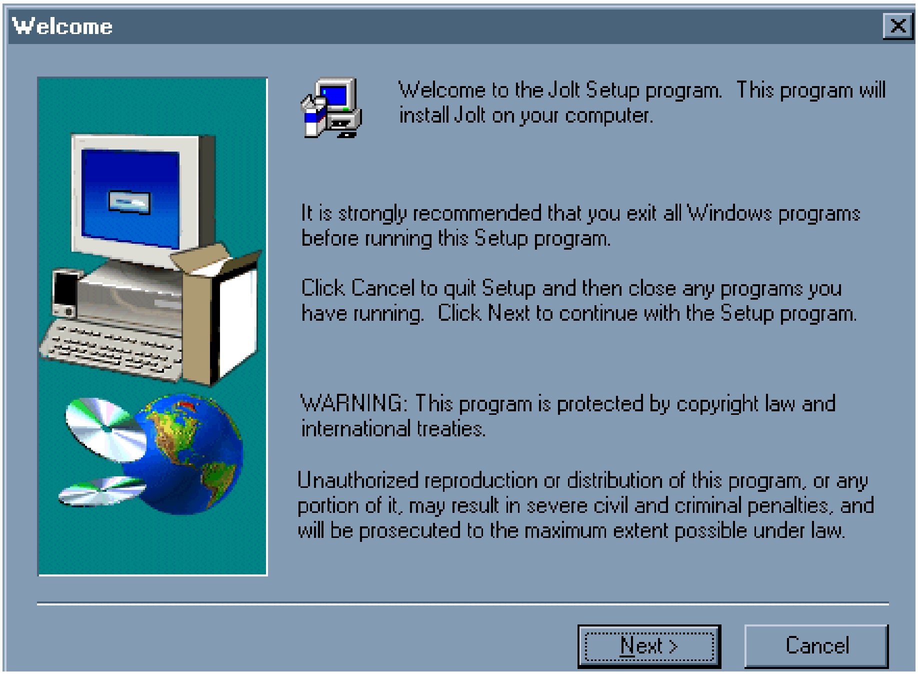
The installation (`setup.exe`) is launched automatically when you insert the Jolt Installation CD and provides a set of step-by-step installation windows to help you quickly install your Jolt product. These windows are self-explanatory.

You can cancel the installation at any time. (If your system detects the presence of a previous Jolt installation, you are given the option of aborting the current installation or overwriting the existing one.)

Refer to Table 1-7 for additional information on the directory structure.

1. When you see the Welcome window shown in Figure 2-1, click the **Next** button to proceed with the installation.

Figure 2-1 Jolt Welcome Window



Welcome to the Jolt Setup program. This program will install Jolt on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

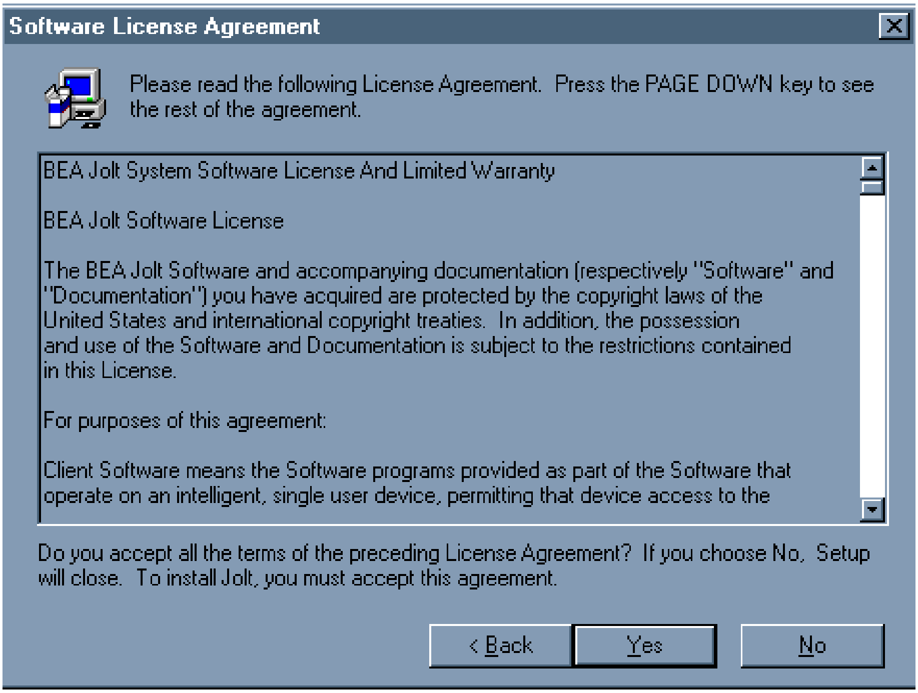
WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

2. The next window that appears is the **Software License Agreement** (Figure 2-2). Use the scroll bar or the **Page Down** key to read the Software License Agreement.

   To continue with the Jolt installation, you must accept the terms of the license agreement. If you accept the terms, click the **Yes** button to continue with the installation. If you do not accept the terms, click the **No** button and the installation stops.

Figure 2-2 Software License Agreement Window



Software License Agreement

Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.

BEA Jolt System Software License And Limited Warranty

BEA Jolt Software License

The BEA Jolt Software and accompanying documentation (respectively "Software" and "Documentation") you have acquired are protected by the copyright laws of the United States and international copyright treaties. In addition, the possession and use of the Software and Documentation is subject to the restrictions contained in this License.

For purposes of this agreement:

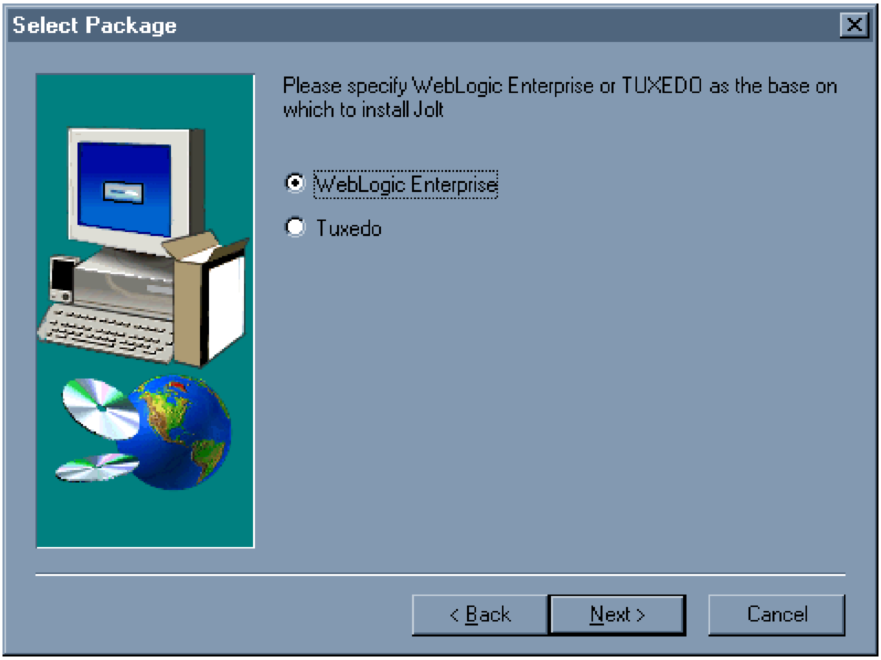
Client Software means the Software programs provided as part of the Software that operate on an intelligent, single user device, permitting that device access to the

Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install Jolt, you must accept this agreement.

3. To install Jolt, you must have either TUXEDO 6.4 or 6.5 or WebLogic Enterprise (WLE) 4.2 already installed. If you have *both* TUXEDO and WLE installed, the **Select Package** window (Figure 2-3) appears.

   Click either the TUXEDO or WebLogic Enterprise radio button to select the base on which to install Jolt.

Figure 2-3 Select Package Window



Select Package

Please specify WebLogic Enterprise or TUXEDO as the base on which to install Jolt

● WebLogic Enterprise

○ Tuxedo

4. If you have either TUXEDO or WLE installed, the **Select Components** window appears (Figure 2-4).

   In the **Select Components** window, select the module(s) you want to install. (You can choose to install any or all components.) The sample window shown in Figure 2-4 displays TUXEDO components. If you install on WLE, the server component listed would show “Jolt Server for WLE” instead of “Jolt Server for Tuxedo.”

Note: Selection in this window works as a toggle. To make your selection, click in the check box to the left of the text representing your choice. The window displays a checkmark to the left of your selection. To deselect a component, click on the checkmark to the left of your choice and the checkmark is removed.

Table 2-1 displays what Jolt installs with each selected component.

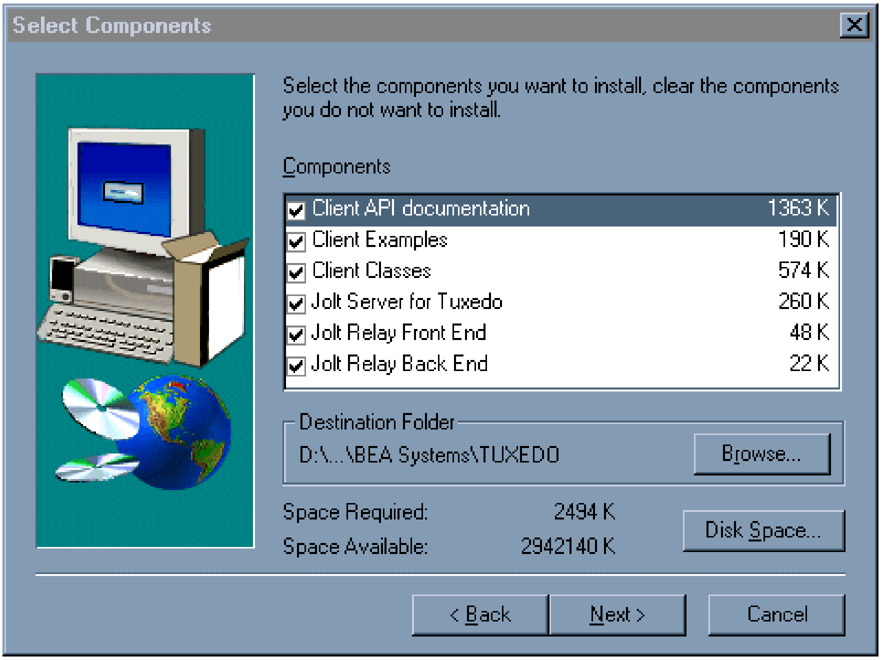
You can check the available and required disk space on a particular drive either in the **Space Available** section at the bottom of the **Select Component** window, or by clicking on **Disk Space** in the same window. The **Space Required** number is approximately equal to the space required by the total number of components you select plus a core component (uninstall files).

The **Destination Folder** displays the directory where the Jolt components are installed. To change the directory, click the **Browse** button and type the directory path. You have another opportunity to specify the Destination Folder when you select the Jolt components for installation (Step 5).

Note: Typically, the destination path is the TUXEDO or WLE directory.

Click the **Next** button to continue with the installation.

Figure 2-4 Select Components Window



Select the components you want to install, clear the components you do not want to install.

Components

- ☑ Client API documentation 1363 K
- ☑ Client Examples 190 K
- ☑ Client Classes 574 K
- ☑ Jolt Server for Tuxedo 260 K
- ☑ Jolt Relay Front End 48 K
- ☑ Jolt Relay Back End 22 K

Destination Folder
D:\...\BEA Systems\TUXEDO

Space Required: 2494 K
Space Available: 2942140 K

Table 2-1 describes what gets installed with each component.

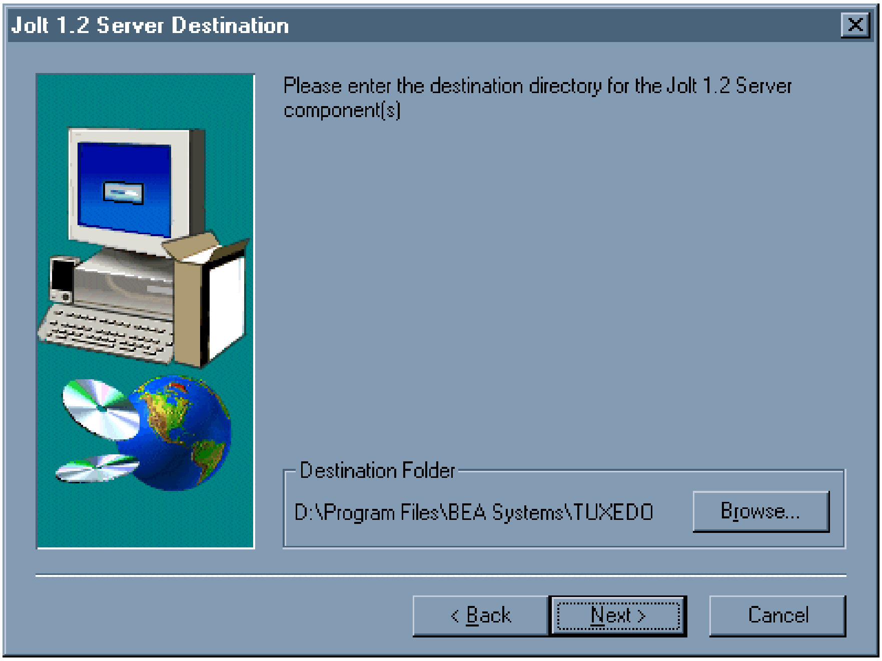
Table 2-1 What Gets Installed with Each Component

Component...	Installs...
Client API Documentation	Javadoc Language Reference documentation only
Client Examples	Client Examples only
Client Classes	Client classes (including JoltBeans) only
Jolt Server for TUXEDO/WLE	Jolt server only (review license)
Jolt Relay Front-End	Jolt Internet Relay front-end server
Jolt Relay Back-End	Jolt Internet Relay back-end server

5. A Component Destination window appears for each component you have chosen to install, that is, each component that you checked in the **Select Component** window. (Figure 2-5 displays the **Server Destination** window.) In each Component Destination window, check that the **Destination Folder** is where you want the component installed. If it is not, click on the **Browse** button and change it.

   Click on the **Next** button.

Figure 2-5 The Server Destination Window



Jolt 1.2 Server Destination

Please enter the destination directory for the Jolt 1.2 Server component(s)

Destination Folder

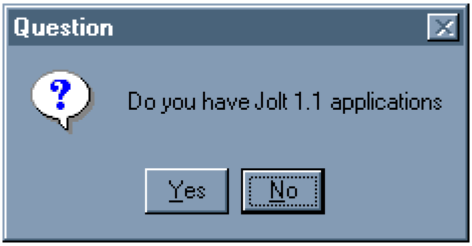
D:\Program Files\BEA Systems\TUXEDO

6. A prompt (Figure 2-6) asks if you have Jolt 1.1 applications.

   If you click the **Yes** button, in addition to the client class `.jar` files, you also get client classes in `.zip` file format and all Jolt client classes in the `<classes/bea/jolt>` directory.

   If you click the **No** button, you get only Jolt client class `.jar` files.

Figure 2-6 Jolt 1.1 Applications



Question

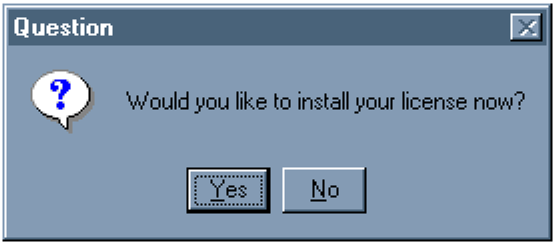
Do you have Jolt 1.1 applications

7. After the selected modules are installed, you are prompted (Figure 2-7) to install your Jolt license.

   If you click the Yes button, the **Insert License Disk** window (Figure 2-8) displays.

   If you click the **No** button, a warning reminds you that your license has not been installed. For further information on licensing, see the "Licensing Your Jolt Software" section later in this chapter.

Figure 2-7 Install License Window



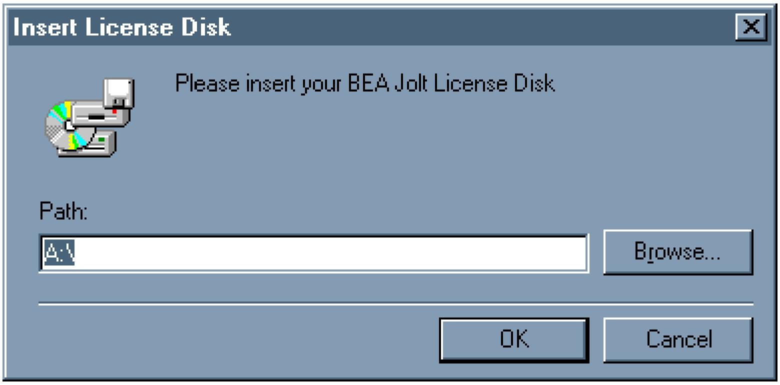
Question

Would you like to install your license now?

8. If you choose to install your license, the **Insert License Disk** window shown in Figure 2-8 displays and you are prompted to insert your license disk.

   Type the designated path or click the **Browse** button to change the drive, then click the **OK** button.

Figure 2-8 Insert License Disk Window



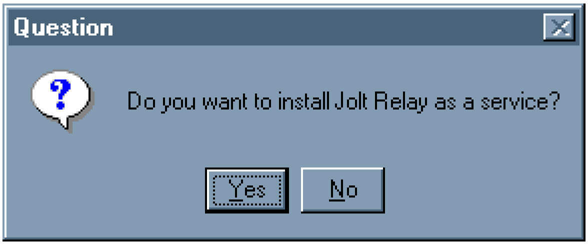
Insert License Disk

Please insert your BEA Jolt License Disk

Path: A:\

9. If you chose to install the Jolt Relay Front End in the **Select Component** window (Figure 2-4), a prompt asks if you want to install Jolt Relay as a service (Figure 2-9). If you click on the **Yes** button, the **Enter Information** (Figure 2-10) window appears. If you click on the **No** button, the **Setup Complete** (Figure 2-11) window appears.

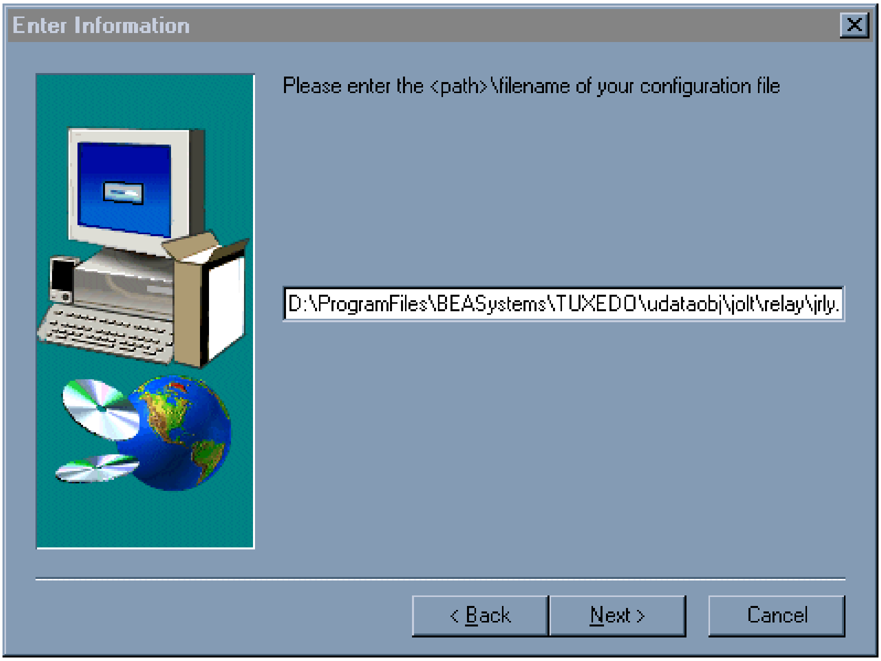
Figure 2-9 Install Jolt Relay Window



Do you want to install Jolt Relay as a service?

10. Enter the path for your configuration file. Click on the **Next** button.

Figure 2-10 The Enter Information Window



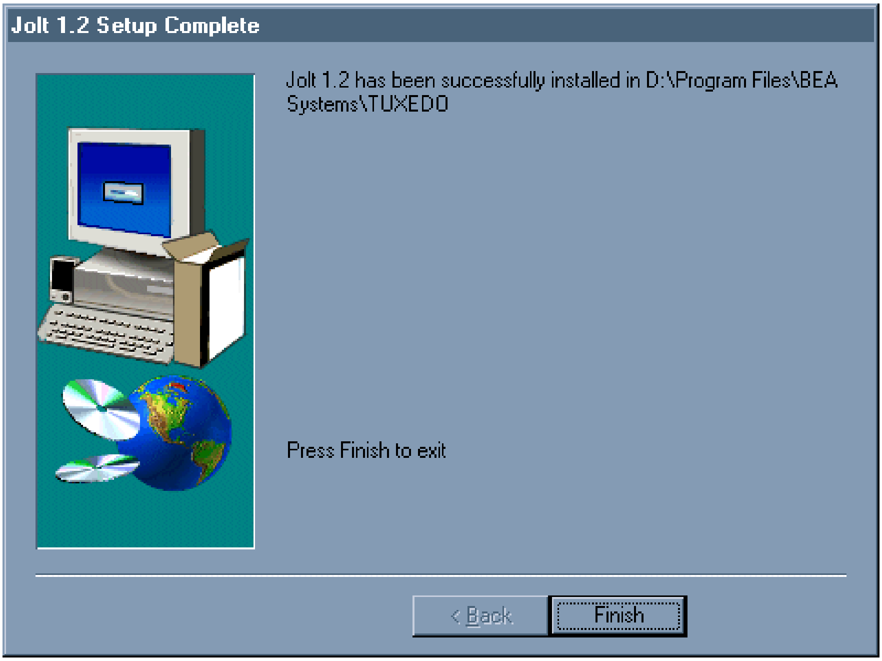
Please enter the <path>\filename of your configuration file

D:\ProgramFiles\BEASystems\TUXEDO\udataobj\jolt\relay\jrly.

11. When all files are installed, the **Setup Complete** (Figure 2-11) window displays.

    Click on **Finish**. Jolt is installed.

Figure 2-11 Setup Complete Window



Jolt 1.2 Setup Complete

Jolt 1.2 has been successfully installed in D:\Program Files\BEA Systems\TUXEDO

Press Finish to exit

UNIX/Linux System Installation Instructions

The Jolt installation shell script for UNIX and Linux systems includes all components necessary for installing the Jolt system: the Jolt Repository, the Jolt Server, the Jolt Relay front-end and back-end, and the Jolt Class Library. Refer to Table 1-6 for an example of the Jolt directory structure.

Invoking the UNIX/Linux Installation Script

Before you install Jolt, make sure that TUXEDO 6.4 or 6.5 or WebLogic Enterprise (WLE) 4.2 has been installed. (If your system detects the presence of a previous Jolt installation, you are given the option of aborting the current installation or overwriting the existing one.)

1. Log in as a user who has write permission in the TUXEDO directory.

2. Insert the CD-ROM in the CD-ROM drive. If you are running on Solaris and the `daemon /usr/sbin/vold` is running, the CD-ROM should be automatically mounted in the `/cdrom/JOLT` directory.

   ```
   cd /cdrom/jolt/unix
   ```

   If you are not running on Solaris or `vold` is not running, consult your UNIX administration documentation to mount the CD-ROM.

3. Type `ls`

   The directory contents should look similar to the following sample. If not, verify that you are installing the correct CD-ROM.

   ```
   hp/ibm/
   install.sh
   seq/sgi/sun5x/ uw/
   ```

4. Type `install.sh`

5. Press **Enter**.

   This invokes the Jolt installation script. The step-by-step install screens are described in the following section.

Unix/Linux System Installation Script

The UNIX system installation script provides a set of step-by-step instructions to help you quickly install your Jolt product. This script lets you specify your platform, operating system, and other installation details. The installation script prompts you through the entire installation process. You can cancel the installation at any time by pressing **CTRL-C** simultaneously.

Note: The script used to show the UNIX installation is taken from Jolt 1.2 for TUXEDO 6.4/6.5 or WLE 4.2.

```
The following installation options are available:

  1     Tuxedo          Install Jolt for Tuxedo

  2     WLE             Install Jolt for WLE

Select an option (default: Tuxedo) [?,??,q]: 1


 01) alpha/dux40/Tux6.5           02) hp/hpux1020/Tux6.4_6.5

 03) hp/hpux11/Tux6.4_6.5         04) ibm/aix42/Tux6.4_6.5

 05) ibm/aix43/Tux6.4_6.5         06) linux/inlnx52/Tux6.4_6.5

 07) seq/dynix442/Tux6.5          08) sgi/irix64/Tux6.4_6.5

 09) sgi/irix65/Tux6.5            10) sni/rel544/Tux6.5

 11) sun5x/isol251/Tux6.4_6.5     12) sun5x/isol26/Tux6.4_6.5

 13) sun5x/isol27/Tux6.5          14) sun5x/sol251/Tux6.4_6.5

 15) sun5x/sol26/Tux6.4_6.5       16) sun5x/sol27/Tux6.5

 17) uw/uw70/Tux6.5


Install which platform's files? [01-17, q to quit, l for list]: 15

BEA Jolt Release 1.2

This directory contains the BEA Jolt System for
```

```
SunOS 5.6 (Solaris 2.6) on SPARC

Is this correct? [y,n,q]: y



To terminate the installation at any time press the interrupt key,
typically <del>, <break>, or <ctrl+c>.

The following packages are available:

  1     jolt            BEA Jolt

Select the package(s) you wish to install (or 'all' to install all
packages) (default: all) [?,??,q]: 1



BEA Jolt  (sparc) Release 1.2
Copyright (c) 1997-1999 BEA Systems, Inc.
Portions * Copyright 1986-1997 RSA Data Security, Inc.
All Rights Reserved.
TUXEDO, Jolt, and WebLogic are registered trademarks of BEA Systems,
Inc.
BEA WebLogic Enterprise is a trademark of BEA Systems, Inc.

The following installation options are available:


 1  all                          Install Jolt server, relay, rad,
                                 and client

 2  server                       Install the server only

 3  relay                        Install the relay front-end only

 4  rad                          Install the relay back-end only

 5  client                       Install the client only

 6  compat                       Install the Jolt 1.1 Client
                                 compatibility classes


Select an option (default: all) [?,??,q]: 1


Note that the jolt server will be installed into an existing TUXEDO
or WLE directory. You MUST have previously installed TUXEDO version
6.4 or 6.5 or WLE 4.2 to attempt this installation.
```

```
Base directory of existing TUXEDO installation [?,q]:
/usr/jolt/T6.4u

Determining if sufficient space is available ...
3068 blocks are required
1045034 blocks are available to /usr/jolt/T6.4u

Using /usr/jolt/T6.4u as the TUXEDO base directory

Unloading /cdrom/cdrom0/unix/sun5x/sol26/Tux6.4_6.5/jolt/joltclt.Z
...
udataobj/jolt/jolt.jar
udataobj/jolt/joltasp.jar
udataobj/jolt/joltjse.jar
udataobj/jolt/JoltBeanDev.jar
udataobj/jolt/JoltBeanDevAwt.jar
udataobj/jolt/JoltBeanDevSwing.jar
udataobj/jolt/JoltBeanDevSwing11.jar
udataobj/jolt/JoltBeanRt.jar
udataobj/jolt/JoltBeanRtAwt.jar
udataobj/jolt/JoltBeanRtSwing.jar
udataobj/jolt/JoltBeanRtSwing11.jar
udataobj/jolt/RE.html
udataobj/jolt/examples/asp/bankapp/templates/deposit.temp
udataobj/jolt/examples/asp/bankapp/templates/inquiry.temp
udataobj/jolt/examples/asp/bankapp/templates/transfer.temp
udataobj/jolt/examples/asp/bankapp/templates/withdrawal.temp
udataobj/jolt/examples/asp/bankapp/templates/error.temp
udataobj/jolt/examples/asp/bankapp/templates/nosession.temp
udataobj/jolt/examples/asp/bankapp/templates/syserror.temp
udataobj/jolt/examples/asp/bankapp/bankapp.properties
udataobj/jolt/examples/asp/bankapp/depositForm.htm
udataobj/jolt/examples/asp/bankapp/inquiryForm.htm
udataobj/jolt/examples/asp/bankapp/transferForm.htm
udataobj/jolt/examples/asp/bankapp/withdrawalForm.htm
udataobj/jolt/examples/asp/bankapp/tellerForm.asp
udataobj/jolt/examples/asp/bankapp/tlr.asp
udataobj/jolt/examples/asp/bankapp/web_templates.inc
udataobj/jolt/examples/asp/bankapp/web_admin.inc
......
udataobj/jolt/classes/bea/jolt/ServiceException.class
udataobj/jolt/classes/bea/jolt/Session.class
udataobj/jolt/classes/bea/jolt/SessionException.class
udataobj/jolt/classes/bea/jolt/SessionState.class
udataobj/jolt/classes/bea/jolt/Transaction.class
udataobj/jolt/classes/bea/jolt/TransactionException.class
udataobj/jolt/classes/bea/jolt/UrlHdlr.class
udataobj/jolt/classes/bea/jolt/UrlInfo.class
udataobj/jolt/jolt.zip
```

```
910 blocks
... finished


Unloading /cdrom/cdrom0/unix/sun5x/sol26/Tux6.4_6.5/jolt/joltrad.Z
...
bin/JRAD
locale/CATNAMES
locale/C/JRAD_CAT
locale/C/JRAD.text
140 blocks


... finished

Unloading /cdrom/cdrom0/unix/sun5x/sol26/Tux6.4_6.5/jolt/joltsvr.Z
...
bin/JREPSVR
bin/JSL
bin/JSH
lib/libjconv.so
include/jotypes.h
locale/CATNAMES
locale/C/JOLT_CAT
locale/C/JOLT.text
udataobj/jrep.f32
udataobj/jwsladmin.f32
udataobj/jolt/repository/jrepository
760 blocks
... finished



Please don't forget to manually append the contents of the license
file
from the enclosed floppy to '/usr/jolt/T6.4u/udataobj/lic.txt'.
Refer to the BEA Jolt manual for details on how to do this.

Changing file permissions...
... finished

Installation of BEA Jolt was successful

The following packages are available:

  1     jolt            BEA Jolt

Select the package(s) you wish to install (or 'all' to install
all packages) (default: all) [?,??,q]: q
```

OS 390 Installation Instructions

To install Jolt on the OS 390 System: (Refer to Table 1-8 for an example of the Jolt directory structure.)

1. Since the OS 390 machine does not have CD-ROM access, you must insert the Jolt CD into a UNIX machine that *does* have CD-ROM access.
2. To access a CD-ROM from an OS 390 machine, use NFS or Samba talking to a UNIX workstation with a CD-ROM drive and mount the CD.
3. Go to the OS 390 machine and use File Transfer Protocol (FTP) to transfer the `tar` file from the OS 390 directory to the OS 390 machine.
4. Use `tar xvf` to untar that file.
5. Now just run `install.sh` to install Jolt.

Note: If you don’t install Jolt on an OS 390 machine in this manner, the installation code is unreadable from the OS 390 machine.

OS 400 Installation Instructions

To Install Jolt on the OS 400 System: (Refer to Table 1-8 for an example of the Jolt directory structure.)

1. Prior to installation, sign on to the OS 400 system and make sure that there is a "tuxedo" user profile (create one if necessary). This user profile is the owner of all the program objects created during the installation process.
2. Find a PC with a CD-ROM drive on the same network as your OS 400, and insert the CD.

   If installing from NT or UNIX, follow Steps 3 and 4.
3. Create a three-line file containing the following File Transfer Protocol (FTP) information:

The open OS 400<*machine-name*>

Account-name

Password

For example, the file for user 'rtl' on the OS 400 machine named 'lc400b.beasys.com' with a password of 'rtl123' would contain the three following lines.

```
open lc400b.beasys.com

rtl

rtl123
```

This file is used to automatically execute the pre-installation procedures necessary for installing Jolt through FTP. Once the pre-installation steps are completed, you can remove the file.

4. Make an entry in your `.netrc` file representing the OS 400 machine on which you are installing Jolt. This entry should include the machine name, your account and password. For example, if the machine name is "lc400b.beasys.com", the account is "rtl" and the password is "rtl123," enter the following information:

   Machine: `lc400b.beasys.com`

   Login: `rtl`

   Password: `rtl123`

   For more information on the`.netrc` file, consult the UNIX System Manual Page for FTP.

5. *(On NT only)* Invoke the pre-installation batch file, `preinst.bat`, with the appropriate command line arguments.

   Usage:

   `preinst.bat` <*CD-drive*> `tmplib tuxdir tuxlib` <*filename*> *version*

   where <*CD-drive*> is the letter that represents the CDROM device on the machine where Jolt is loaded. For example `D:tmpl` is the library on the OS400 where the Jolt save files will be copied from the CDROM.

   `tuxdir`

   is the IFS file system on the OS400 where TUXEDO is already installed and where the Jolt files will be installed.

`tuxlib`

is the TUXEDO library on the OS400 where the save files will be restored.

`<filename>`

is the full path to a file that contains FTP account and password information for the OS 400 machine where Jolt will be installed.

`version`

is the Jolt version (either 'us' or 'intl') to be installed.

NOTE: If you have problems with the `preinst.bat` from the CD, copy it to your C: drive and repeat the previous instructions.

6. (*On UNIX only*) Invoke the pre-installation shell, `preinst.sh` with the appropriate command line arguments.

   Usage: `preinst.sh mount-point tmplib tuxdir tuxlib` *machine version* where:

   CD-drive

   is the file system on which the Jolt CDROM is mounted, for example `/cdrom`.

   `tmplib`

   is the library on the OS 400 where the Jolt save files will be copied to from the CDROM.

   `tuxdir`

   is the IFS file system on the OS 400 where TUXEDO is already installed and where the Jolt files will be installed.

   `tuxlib`

   is the TUXEDO library on the OS 400 where the save files will be restored.

   machine-name

   is the name of the OS 400 machine on which Jolt is to be installed.

   version

   is the Jolt version (either 'us' or 'intl') to be installed.

7. Run the install program on the OS 400:

```
addlible TUXLIB
```

If the version is US then call:

```
joltinstus parm('TMPLIB''TUXLIB''TUXDIR''all|clt|doc|svr|rad')
```

If the version is International, call:

```
joltinstin parm('TMPLIB' 'TUXLIB''TUXDIR''all|clt|doc|svr|rad')
```

where:

`all` is used to install all Jolt packages,

`clt` is used to install Jolt client only,

`doc` is used to install Jolt doc only,

`svr` is used to install Jolt server only,

`rad` is used to install Jolt rad only.

Note: `all`, `clt`, `doc`, `svr`, or `rad` must be lowercase.

During installation, a number of "restoring X to Y" messages are displayed, for example, "Restoring service programs to `TUXLIB`". The installation process may take several minutes.

8. Remove all `save` files created in Step 5 from `TMPLIB`.

9. Remove `TMPLIB` if it is different from `TUXLIB` and is no longer needed.

10. Your Jolt installation is now complete. Jolt is ready for use as soon as you install a license file in `$TUXDIR/udataobj/lic.txt`.

    For information on obtaining a license, contact your Jolt sales representative.

Before you develop your TUXEDO application:

♦ Make sure your OS 400 is running with the latest CUM tape.In particular, make sure the following PTFs have been applied to your system:

```
sf32801 product 5716ss1

mf12328 product 5716999
```

Use the "`go ptf`" option 5 to view the current PTFs. If necessary, these PTFs can be downloaded from IBM support over the ECS line.

♦ It is highly recommended that you obtain the Client Access software from IBM.

At present, IBM ships Client Access for both Windows 95 and NT. Client Access allows, among other things, IFS directories to be mapped as network drives on your PC. Since TUXEDO writes ULOG files in IFS, this is a convenient way to view them from your PC.

- Make sure TUXLIB is in your library list and proper environment variables such as TUXDIR and TUXCONFIG are set via the setenv or putenv utilities.
- If this is your first encounter with TUXEDO, it is recommended that you consult the sample applications in $TUXDIR/apps, such as simpapp.

Tips for running the Web-based Administration Console on the OS 400:

The Administration Console is new to Release 6.4 and it allows the administration of TUXEDO applications from any Web browser (for example, Netscape or Internet Explorer).

- Two programs, tuxwsvr and wlisten, must be run in the background.

  You can accomplish this by running a command-line script using SBMDBJOB. For example, the command-line script used to start tuxwsvr would look like:

```
//BCHJOB JOB(WEBSERVER) JOBD(TUXADM/JOBD)
addlible TUXLIB
cd /home/tuxadm/web
call setenv env
call tuxwsvr ('-l//lc400b.beasys.com:3184' '-iwinput.ini')
```

- The first line should begin with //BCHJOB, which allows a number of parameters to be set. (See the CL Programming Manual.)

  The JOB parameter, for example, sets the name of the job such that it appears on a WRKACTJOB screen. A customized Job Description is needed here also so that the job can be run in batch. The default Job Description QDFTJOBD for interactive sessions cannot be used here. To create a new one, use WRKJOBD and copy QDFTJOBD to TUXADM/JOBD, then change TUXADM/JOBD by adding a USER name.

- Several environment variables also need to be set, for example, TUXDIR.

  This is done by putting all necessary environment settings in an env file and calling setenv on it. Other application-specific variables, for example APPDIR, can be set here as well.

- One additional input file that tuxwsvr expects is an INI file.

This INI file tells the server where to look for HTML pages and Java images. For example:

```
CGI /cgi-bin <TUXDIR>/bin

HTML /java/images <TUXDIR>/web/gui/java/bea/tuxadm/images

JAVA /java <TUXDIR>/web/gui/java

HTML /docs <TUXDIR>/docs

HTML / <TUXDIR>/udataobj/webgui
```

- Substituting the actual value of the TUXEDO root directory in winput.ini (for example, if /qopensys/tuxr64 was where TUXEDO was installed) for <TUXDIR> would yield the following INI file:

```
CGI /cgi-bin /qopensys/tuxr64/bin

HTML/java/images/qopensys/tuxr64/web/gui/java/bea/tuxadm/images

JAVA /java /qopensys/tuxr64/web/gui/java

HTML /docs /qopensys/tuxr64/docs

HTML /qopensys/tuxr64/udataobj/webgui
```

- For more detailed information on creating the tuxwsvr initialization file, see the tuxwsvr manual page in section 1 of the TUXEDO Reference Manuals.
- Current implementation also requires that for all user CGI programs, the HFS library in which they reside be added to the library list.
- The batch job setup for wlisten is similar to that of tuxwsvr.
- All wlisten options are taken from an initialization file that is specified by the -i option.

  This INI file must be created. It is *not* generated during installation on the OS400. For detailed information on its creation, consult the wlisten manual page in Section 1 of the *TUXEDO Reference Manual*.

  An example is provided below:

```
TUXDIR=/qopensys/tuxr64

NADDR=0x00020c71c604a68d

CODEBASE=/java

DOCBASE=http://lc400b:3184/docs
```

To start the Administration Console on the OS 400:

Open the following URL:

`http://<`*`machine_name`*`>:<`*`port`*`>/cgi-bin/tuxadm/TUXDIR=<TUXDIR>&INIFILE=<inifile>`

where:

- `<TUXDIR>` is the TUXEDO installation directory specified during the installation
- `<inifile>` refers to the initialization file for `wlisten`, `<machine_name>`
- `:<`*`port`*`>` is the OS 400 machine/port combination on which `tuxwsvr` is running.

For additional information on the Administration Console setup, consult the *TUXEDO Installation Guide*.

Licensing Jolt for TUXEDO 6.4 and 6.5

The licensing scheme used by Jolt depends on the version of TUXEDO used with Jolt. You need the following information to install your license:

- The version of TUXEDO you are installing.
- Your TUXEDO directory (`TUXDIR`) from the installation. This is the directory that contains the TUXEDO directories (`bin`, `udataobj`, etc.).
- Your license file

Jolt for TUXEDO 6.4 and 6.5 uses a digitally signed license file to enable a license. This file is provided on a floppy disk shipped with your Jolt software.

The UNIX installation program (`install.sh`) does not install the license automatically; see the “UNIX Licensing Instructions” in the following section.

The NT installation program (`setup.exe`) prompts you for the location of the Jolt license file. When you provide the necessary information, the installation program installs the license file for you. If you do not install the license file during installation, follow the steps in the “NT Licensing Instructions” section.

UNIX Licensing Instructions

1. Identify your current TUXEDO license file. This is located in `$TUXDIR/udataobj/lic.txt.`
2. Make a copy of this file:

   ```
   cd $TUXDIR/udataobj
   cp lic.txt lic.txt.bak
   ```

3. Check that you have completed Step 2. Verify the copy using OS-specific commands (e.g., `diff` on UNIX systems).
4. Append the contents of the Jolt license file to the TUXEDO license file:

   ```
   cat /dev/diskette/joltlic.txt >> lic.txt
   ```

NT Licensing Instructions

1. Identify your current TUXEDO license file. This is located in `%TUXDIR%\udataobj\lic.txt`.
2. Make a copy of this file:

   ```
   cd %TUXDIR%\udataobj
   copy lic.txt lic.txt.bak
   ```

3. Check that you have a completed Step 2. Verify the copy using OS-specific commands.
4. Append the contents of the Jolt license file to the TUXEDO license file:

   ```
   copy lic.txt + a:\joltlic.txt
   ```

   A text editor can be used to copy and paste the contents of the Jolt license file into the TUXEDO license file.

   Note: The digital signature is 64 characters long. Every character must match exactly or the license is not valid.

Installing Jolt Online Documentation

Accompanying your Jolt software is an online, HTML-based, documentation set to assist you with using Jolt. The Jolt product CD-ROM contains the HTML version of the *BEA Jolt Installation and Configuration Guide* and the *BEA Jolt Developer's Guide*.

CHAPTER

3 Configuring the Jolt System

This chapter explains how to configure Jolt. It contains a "Quick Configuration" section for users who are familiar with Jolt. The rest of the chapter provides more detailed information. It is presumed that readers of this chapter are system administrators or application developers who have experience with the operating systems and workstation platforms on which they are configuring Jolt.

This chapter includes the following sections:

- Quick Configuration
- Jolt Background Information
- Jolt Relay
- Jolt Relay Adapter
- Jolt Repository
- Event Subscription
- TUXEDO Background Information
- Sample Applications in Jolt Online Resources

Quick Configuration

If you are already familiar with Jolt and TUXEDO, this section provides a quick guide to the configuration procedure. If you have not used Jolt before, read the “Jolt Background Information” section in this chapter.

Configure Jolt on TUXEDO

Follow the directions in this section to configure the Jolt Server Listener (JSL).

Edit the UBBCONFIG file

1. In the `MACHINES` Section, specify `MAXWSCLIENTS=`*`number`* (Required).

Note: If `MAXWSCLIENTS` is not set, JSL does not boot.

2. In the `GROUPS` section, set `GROUPNAME` <*`required parameters`*> [*`optional parameters`*].
3. Set the `SERVERS` section (Required).

   Lines within this section have the form:

   `JSL` <*`required parameters`*> [*`optional parameters`*]

   where `JSL` specifies the file (*`string_value`*) to be executed by `tmboot(1)`.
4. Set the required parameters for `JSL`.

   Required parameters are:

   `SVRGRP`, `SVRID`, and `CLOPT="-A...-n...//`*<host port>*`"`
5. Set other parameters for `JSL`.

   The following parameters can be used with the JSL, but you need to understand how doing so would affect your application. See “Other Parameters Usable With JSL” later in this chapter.

   `SVRGRP=`*`string_value`*

   `SRVID=`*`number`*

MAX # of JSHs

MIN # of JSHs

Configure the Jolt Repository

In the `GROUPS` section:

1. Specify the same identifiers given as the value of the `LMID` parameter in the `MACHINES` section.
2. Specify the value of the `GRPNO,` between 1 and 30,000.

In the `SERVERS` section:

The Jolt Repository Server (`JREPSVR`) contains services for accessing and editing the Repository. Multiple `JREPSVR` instances share repository information through a shared file. Include `JREPSVR` in the `SERVERS` section of the `UBBCONFIG` file.

1. Indicate a new server identification with the `SRVID` parameter.
2. Specify the `-W` flag for one (and only one) `JREPSVR` to ensure that you can edit the Repository. (Without this flag, the Repository is read-only.)
3. Type the `-P` flag to specify the path of the repository file. (An error message displays in the TUXEDO `ULOG` file if the argument for the `-P` flag is not entered.)
4. Add the file pathname of the Repository file (for example, `/app/jrepository`).
5. Boot the TUXEDO system using the `tmloadcf` and `tmboot` commands.

Initialize Services Using TUXEDO and the Repository Editor

You must initially define the TUXEDO services using TUXEDO and Jolt in order to make the Jolt services available to the client.

1. Build the TUXEDO server that contains the service.
2. Access the Jolt Repository Editor.

Before You start the Repository Editor

Before you start the Repository Editor, make sure that you have installed all necessary Jolt software. You cannot use the Repository Editor until `JREPSVR` and `JSL` are running.

Start the Repository Editor from either the JavaSoft `appletviewer` or from your Web browser.

Starting the Repository Editor Using appletviewer:

1. Set the CLASSPATH to include the Jolt class directory or the directory where the `*.jar` files reside.
2. If loading the applet from a local disk, type the following at the URL location:

   `appletviewer <`*full-pathname*`>/RE.html`

   If loading the applet from the Web server, type the following at the URL location:

   `http://<www.server>/<URL path>/RE.html`
3. Press **Enter**. The Repository Editor logon window displays.

Starting the Repository Editor from a Web server:

To start the Repository Editor from a local file:

1. Set the CLASSPATH to include the Jolt class directory.
2. Type: `file:<`*full-pathname*`>/RE.html`

To start the Repository Editor from a Web server:

1. Ensure that the `CLASSPATH` does not include the Jolt class directory
2. Remove the Jolt classes from `CLASSPATH`.
3. Type the following:

   `http://<www.server>/<URL path>/RE.html`

Note: If `jolt.jar` and `admin.jar` are in the same directory as `RE.html`, the web server provides the classes. If they are not in the same directory as `RE.html`, modify the applet code base.

4. Press **Enter**. The Repository Editor logon window displays.

Log on to the Repository Editor

After starting the Jolt Repository Editor, follow these directions to log on:

1. Type the name of the server machine designated as the “access point” to the TUXEDO application and select the port number text field.
2. Type the port number and press **Enter**. The system validates the server and port information.

   Note: Unless you are logging on through the Jolt Relay, the same port number is used to configure the Jolt Listener. Refer to your `UBBCONFIG` file for additional information.

3. Type the TUXEDO Application Password and press **Enter**. Based on the authentication level, type the remaining information.
4. Type the TUXEDO user name and press **Tab**.
5. Type the TUXEDO user password and press **Enter**.

Note: The Jolt 1.2 Repository Editor uses the hardcoded `joltadmin` for the user role.

The **Packages** and **Services** options are activated.

Repository Editor Logon Window Description

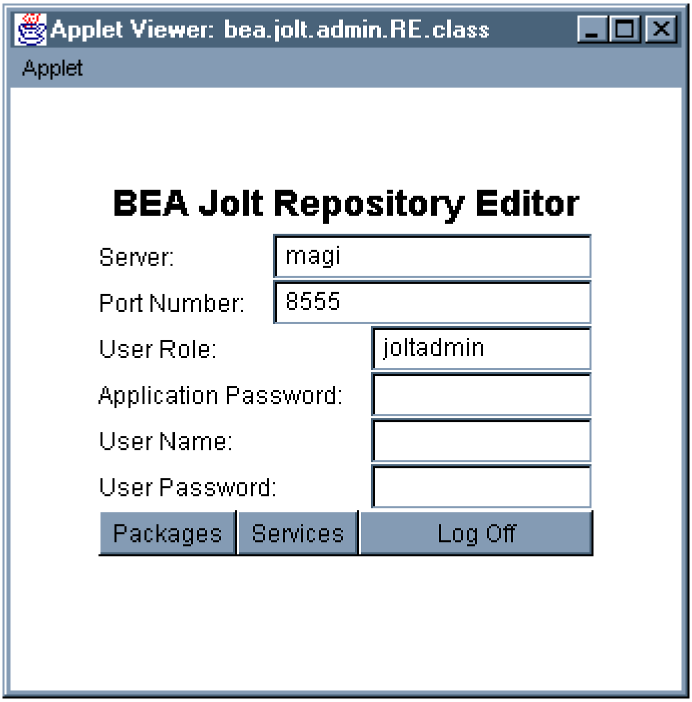
The following table details the options of the Repository Editor logon window.

Option	Description
Server	Type the server name.
Port Number	Type the port number in decimal value. **Note:** After the server name and port number are entered, the user name and password fields are activated. Activation is based on the authentication level of the TUXEDO application.
Application Password	TUXEDO administrative password text entry.
User Name	TUXEDO user identification text entry. The first character must be an alpha character.
User Password	TUXEDO password text entry.
User Role	TUXEDO user role. Required only if TUXEDO authentication level is USER_AUTH or higher.
Packages	Accesses the Packages window. (Enabled after the logon.)
Services	Accesses the Services window. (Enabled after the logon.)

Exit the Repository Editor

Exit the Repository Editor when you are finished adding, editing, testing, or deleting packages, services, and parameters. Figure 3-1 is an example of the Repository Editor window before exiting. Only **Packages**, **Services**, and **Close** are enabled. All text entry fields are disabled.

Figure 3-1 Example of the Repository Editor Logon Window Before Exiting



BEA Jolt Repository Editor

Server: magi

Port Number: 8555

User Role: joltadmin

Application Password:

User Name:

User Password:

To exit the Repository Editor:

1. Select **Back** from a previous window to return to the Logon window.
2. Select **Close** to terminate the connection with the server. The Repository Editor Logon window continues to display with disabled fields.
3. Select **Close** from your browser menu to remove the window from your screen.

Configure the TUXEDO TMUSREVT Server for Event Subscription

Jolt Event Subscription is used to receive event notifications from either TUXEDO services or other TUXEDO clients. Configure the TUXEDO `TMUSREVT` server and modify the application `UBBCONFIG` file. Listing 3-1 shows the relevant `TMUSREVT` parameters in the `UBBCONFIG` File:

Listing 3-1 TMUSREVT Parameters in the UBBCONFIG File

```
TMUSREVT        SRVGRP=EVBGRP1  SRVID=40        GRACE=3600
                ENVFILE="/usr/tuxedo/bankapp/TMUSREVT.ENV"
                CLOPT="-e tmusrevt.out -o tmusrevt.out -A --
                -f /usr/tuxedo/bankapp/tmusrevt.dat"
                SEQUENCE=11
```

In the `SERVERS` sections of the `UBBCONFIG` file, specify the `SRVGRP and SRVID.`

Configure Jolt Relay

On UNIX:

Start the JRLY process on UNIX by typing the following command a the system prompt:

```
jrly -f <config_file_path>
```

If the configuration file does not exist or cannot be opened, the JRLY writes a message to standard error, attempts to log the startup failure in the error log, then exits.

On UNIX and NT:

The format of the configuration file is a TAG=VALUE format. Blank lines or lines starting with a "#" are ignored. Listing 3-2 is an example of the formal specifications of the configuration file.

Listing 3-2 Formal Configuration File Specifications

```
LOGDIR=<LOG_DIRECTORY_PATH>
ACCESS_LOG=<ACCESS_FILE_NAME in LOGDIR>
ERROR_LOG=<ERROR_FILE_NAME in LOGDIR>
LISTEN=<IP:Port combination where JRLY will accept
comma-separated connections>
CONNECT=<IP:Port1, IP:Port2...IP:PortN:Port(List of IP:Port
combinations associated with JRADs: can be 1...N)>
```

On NT Only (Optional):

SOCKETTIMEOUT is the time in seconds for which JRLY NT service blocks for network activity (new connections, data to be read, closed connections). SOCKETTIMEOUT also affects the SCM. When the SCM requests the NT service to stop, the SCM must wait for at least SOCKETTIMEOUT seconds before quitting.

Note: The format for directory and file names is determined by the operating system. UNIX systems use the forward slash (/). NT systems use the backslash (\). If any of the files specified in LOGDIR, ACCESS_LOG, or ERROR_LOG cannot be opened for writing, the JRLY prints an error message on stderr and exits.

The format for host names and port numbers are shown in Table 3-1.

Table 3-1 Host Name and Port Number Formats

Host Name/Port Number	Description
Hostname:Port	Hostname is a string; Port is a decimal number.
//Hostname:Port	Hostname is a string; Port is a decimal number.
IP:Port	IP is a dotted notation IP address; Port is a decimal number.

Start the Jolt Relay Adapter (JRAD).

1. Type: `tmloadcf -y <UBBFILE>`
2. Type: `tmboot`

Configure the JRAD

A single JRAD process can only be connected to a single JRLY. A JRAD can be configured to communicate with only one JSL and its associated JSH. However, multiple JRADs can be configured to communicate with one JSL. The CLOPT parameter for TUXEDO services must be included in the UBBCONFIG file.

1. Type: `-l <hexadecimal format>` (The port to the JSL for the JRLY to connect on behalf of the client.)
2. Type: `-c <hexadecimal format>` (The address of the corresponding JSL to which JRAD connects.)

Note: The format is 0x0002PPPNNN or, in dot notation, 100.100.10.100.

3. Configure networked components. **Jolt is now configured.**

Jolt Background Information

This section contains additional information on Jolt components.

Jolt Server

The Jolt Server is a listener that supports one or more handlers.

Jolt Server Listener (JSL). The JSL is configured to support clients on an IP/port combination.The JSL works with the Jolt Server Handler (JSH) to provide client connectivity to the backend of the Jolt system. The JSL runs as a TUXEDO server.

Jolt Server Handler (JSH). The JSH is a program that runs on a TUXEDO server machine to provide a network connection point for remote clients. The JSH works with the JSL to provide client connectivity residing on the backend of the Jolt system. More than one JSH can be available to the JSL, up to 32,767. (See description of the `-M` command-line option on page 3-26.)

System Administrator Responsibilities. The system administrator's responsibilities for the server components of Jolt include:

- Determining the JSL network address.
- Determining the number of Jolt clients to be serviced. (The number of clients to be serviced is limited by `MAXWSCLIENTS` in `UBB`.)
- Determining the minimum and maximum number of JSHs.

Starting the JSL

To start all administrative and server processes in the `UBBCONFIG` file:

1. Type: `tmloadcf`

   This command parses the configuration file and loads the binary version of the configuration file.

2. Type: `tmboot -y`

   This command activates the application specified in the configuration file.

If you do not enter any options, a prompt asks you if you really want to overwrite your `TUXCONFIG` file.

See the *Administering the BEA TUXEDO System* or the *BEA TUXEDO Reference Manual* for information on `tmloadcf` and `tmboot`.

Shutting Down the JSL

All shutdown requests to the Jolt servers are initiated by the TUXEDO command:

```
tmshutdown -y.
```

During shutdown:

- No new client connections are accepted.
- All current client connections are terminated. TUXEDO rolls back in-flight transactions. Each client receives an error message indicating that the service is unavailable.

Restarting the JSL

TUXEDO monitors the JSL and restarts it in the event of a failure. When TUXEDO restarts the listener process, the following occurs:

- Clients attempting a listener connection must try to reconnect. Clients attempting a handler connection receive a timeout or a time delay.
- Clients currently connected to a handler are disconnected (JSH exits when its corresponding JSL exits normally).

Configuring the JSL

The Jolt Server Listener (JSL) is a BEA TUXEDO server responsible for distributing connection requests from Jolt to the Jolt Server Handler (JSH). BEA TUXEDO must be running on the host machine where the JSL and JREPSVR are located.

JSL Command-Line Options

The server may need to obtain information from the command line. The CLOPT parameter allows you to specify command-line options that can change some defaults in the server. The JSL command-line options are explained in Table 3-2.

Table 3-2 JSL Command-Line Options

JSL Command Line Option	Description
`[-c` *`compression_threshold`*`]`	Enables application data sent between a Jolt client and a Jolt server (JSH) to be compressed during transmission over the network. *`compression_threshold`* is a number that you specify between 0 and 2,147,483,647 bytes. Any messages that are larger than the specified compression threshold are compressed before transmission. The default is no compression; that is, if no compression threshold is specified, Jolt does not compress messages on client or server. **Note:** The previous `-c` *`connection-mode`* option has been replaced with the `-j` *`connection-mode`* option.
`[-d` *`device_name`*`]`	The device for platforms using the Transport Layer Interface. There is no default. Required. (Optional for sockets)

Table 3-2 JSL Command-Line Options

JSL Command Line Option	Description
`[-H` *`external netaddr`*`]`	*`external netaddr`* is the network address Jolt clients use to connect to the application. The JSL process uses this address to listen for clients attempting to connect at this address. If the address is `0x0002MMMMdddddddd` and JSH network address is `0x00021111ffffffff`, the known network address is `0x00021111dddd dddd`. If the address starts with "//" network address, the type is IP based and the TCP/IP port number of JSH network address is copied into the address to form the combined network address. The IP address must be specified in the following form: `-H //external ip address:MMMM` (Optional for JSL in TUXEDO 6.4 and 6.5)
`[-I` *`init-timeout`*`]`	The time (in seconds) that a Jolt client is allowed to complete initialization through the JSH before it is timed out by the JSL. Default is 60 seconds. (Optional)
`[-j` *`connection_mode`*`]`	The following connection modes from clients are allowed: RETAINED - the network connection is retained for the full duration of a session. RECONNECT - the client establishes and brings down a connection when an idle timeout is reached, reconnecting for multiple requests within a session. ANY - the server allows a client to request either a RETAINED or RECONNECT type of connection for a session. The default is ANY. That is, if no option is specified, the server allows a client to request either a RETAINED or RECONNECT type of connection. (Optional) **Note:** This option has been changed in this release from `-c` *`[connection_mode]`* to `-j` *`[connection_mode]`*.
`[-m` *`minh`*`]`	The minimum number of JSHs that are available in conjunction with the JSL at one time. The range of this parameter is from 0 through 255. Default is 0. (Optional)

Table 3-2 JSL Command-Line Options

JSL Command Line Option	Description
`[-M` *`maxh`*`]`	The maximum number of JSHs that are available in conjunction with the JSL at one time. If this option is not specified, the parameter defaults to `MAXWSCLIENTS` divided by the rounded-up `-x` multiplexing factor (MPX). If specified, the `-M` option takes a value from 1 to 32,767. (Optional)

`[-n` *`netaddr`*`]`	Network address used by the Jolt listener with TUXEDO 6.4 and 6.5, and WLE 4.2.
	TCP/IP addresses may be specified in the following formats:
	`"//host.name:port_number"`
	`"//#.#.#.#:port_number"`
	In the first format, the domain finds an address for hostname using the local name resolution facilities (usually DNS). Hostname must be the local machine, and the local name resolution facilities must unambiguously resolve hostname to the address of the local machine.
	In the second example, the "#.#.#.#" is in dotted decimal format. In dotted decimal format, each # should be a number from 0 to 255. This dotted decimal number represents the IP address of the local machine.

Table 3-2 JSL Command-Line Options

JSL Command Line Option	Description
	In both of the above formats, `port_number` is the TCP port number at which the domain process listens for incoming requests. `port_number` can either be a number between 0 and 65535 or a name. If `port_number` is a name, then it must be found in the network services database on your local machine. The address can also be specified in hexadecimal format when preceded by the characters “0x”. Each character after the initial “0x” is a number from 0 to 9 or a letter from A to F (case insensitive). The hexadecimal format is useful for arbitrary binary network addresses such as IPX/SPX or TCP/IP. There is no default. (Required)
`[-T` *`Client-timeout`*`]`	The time (in minutes) allowed for a client to stay idle. If a client does not make any requests during this time, the JSH disconnects the client and the session is terminated. If an argument is not supplied, the session does not timeout. When the `-j ANY` or `-j RECONNECT` option is used, always specify `-T` with an idle timeout value. If `-T` is not specified and the connection is suspended, JSH does not automatically terminate the session. The session never terminates if a client abnormally ends the session. If a parameter is not specified, the default is no timeout. (Optional)
`[-w` *`JSH`*`]`	This command line option indicates the Jolt Server Handler. Default is JSH. (Optional)
`[-x` *`mpx-factor`*`]`	This is the number of clients that one JSH can service. Use this parameter to control the degree of multiplexing within each JSH process. If specified, this parameter takes a value from 1 to 32767 for UNIX and NT. Default value is 10. (Optional)

Table 3-2 JSL Command-Line Options

JSL Command Line Option	Description
`[-Z 0|40|128]`	When establishing a network link between a Jolt client and the JSH, allows encryption up to this level.The initial 0 means no DH nodes, no RC4. 40 and 128 specify the length (in bits) of the encryption key. DH key exchange is needed to generate keys. Session keys are not transmitted over the network. The default value is 0.

Security and Encryption

Authentication and key exchange data are transmitted between Jolt clients and the JSL/JSH using the Diffie-Hellman key exchange. All subsequent exchanges are encrypted using RC4 encryption. International packages use a DES key exchange and a 128-bit key, with 40 bits encrypted and 88 bits exposed.

Programs using the 128-bit encryption cannot be exported outside the United States without proper approval from the United States government. Customers with intranets extending beyond the United States cannot use this mode of encryption if any internal clients are outside the United States.

Jolt Relay

The combination of the Jolt Relay (JRLY) and its associated Jolt Relay Adapter (JRAD) is typically referred to as the Internet Relay. Jolt Relay is a component that routes messages from a Jolt client to a JSL or JSH. This eliminates the need for the JSH and TUXEDO to run on the same machine as the Web server (generally considered as insecure). The Jolt Relay consists of two components illustrated in Figure 3-2.

- **Jolt Relay (JRLY)**. The JRLY is the Jolt Relay front-end. It is not a TUXEDO client or server and is not dependent on the TUXEDO version. It is a stand-alone software component. It requires only minimal configuration to allow it to work with Jolt clients.
- **Jolt Relay Adapter (JRAD)**. The JRAD is the Jolt Relay back-end. It is a TUXEDO system server, but does not include any TUXEDO services. It requires command line arguments to allow it to work with the JSL and the TUXEDO system.

Note: The Jolt Relay is transparent to Jolt clients and Jolt servers. A Jolt server can simultaneously connect to intranet clients directly, or via the Jolt Relay to Internet clients.

Figure 3-2 Jolt Internet Relay Path

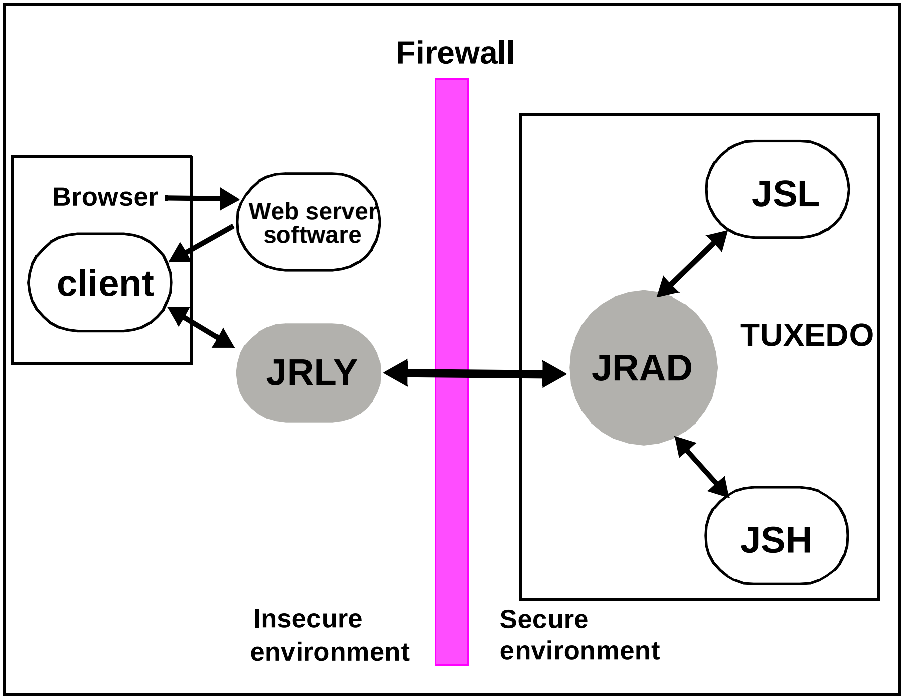


Figure 3-2 shows how a browser connects to the Web server software and downloads the Jolt applets. The Jolt applet or client connects to the JRLY on the Web server machine. The JRLY forwards the Jolt messages across the firewall to the JRAD. The JRAD selectively forwards messages to the JSL or appropriate JSH.

Jolt Relay Failover

There are two points of failovers associated with JRLY:

- Jolt Client to JRLY connection failover
- JRLY to JRAD connection failover

Jolt Client to JRLY Connection Failover

If one server address does not result in a successful session, the failover function allows the Jolt Client API to connect to the next free (unconnected) JRLY specified in the argument list of the API. To enable this failover in an NT environment, multiple NT JRLY services can be executed. In a non-NT environment, multiple JRLY processes are executed. Each JRLY (service or process) has its own configuration file. This type of failover is handled by Jolt 1.2 client API changes that allow you to specify a list of Jolt server addresses (JSL or JRLY).

JRLY to JRAD Adapter Connection Failover

Each JRLY configuration file has a list of JRAD addresses. When a JRAD is unavailable, JRLY tries to connect to the next free (unconnected) JRAD, in a round-robin fashion. Two JRLYs cannot connect to the same JRAD. Given these facts, you can make the connection efficient by giving different JRAD address orders. That is, if you make one extra JRAD available on standby, the first JRLY that loses its JRAD connects to the extra JRAD. This type of failover is handled by JRLY alone.

If any of the listed JRADs are not executing when JRLY is started, the initial connection fails. When a Jolt client tries to connect to JRLY, the JRLY again tries to connect to the JRAD.

To accommodate the failover functionality, you have to boot multiple JRADs by configuring them in the `UBBCONFIG` file.

Jolt Relay Process

The JRLY (front-end relay) process can be started before or after the JRAD is started. If the JRAD is not available when the JRLY is started, the JRLY attempts to connect to the JRAD when it receives a client request. If JRLY is still unable to connect to the JRAD, the client is denied access and a warning is written to the JRLY error log file.

Starting the JRLY on UNIX

Start the JRLY process by typing the command name at a system prompt.

```
jrly -f <config_file_path>
```

If the configuration file does not exist or cannot be opened, the JRLY prints an error message. Refer to Appendix B of the *BEA Jolt Developer's Guide* for the Jolt Relay error messages.

If the JRLY is unable to start, it writes a message to standard error and attempts to log the startup failure in the error log, then exits.

JRLY Command-Line Options for NT

This section discusses command-line options that are available from the NT version of `JRLY.exe`. Note that:

- JRLY as an NT service is available only for Microsoft Windows NT.
- When the display suffix is optional (when `[`*`display_suffix`*`]` is shown), all operations are performed on the default JRLY NT service instance.
- For manually-installed, additional JRLY services, a suffix (any string) is required. Also, you can install the default service manually by omitting the optional string suffix.
- Each instance of JRLY NT service uses the same binary executable file.
- A new process is started for each instance of JRLY NT service.
- The syntax for these options is: `jrly -command.`
- Text specified within brackets (`[ ]`) is optional.
- All the following commands except `-start` and `-stop` require that you have write access to Windows NT Registry.
- The `-start` and `-stop` commands require that you have NT Service control access. These restrictions are based on NT user restrictions.

The command line options are as follows:

`jrly -install [`*`display_suffix`*`]`

Install `jrly` as an NT service.

Example 1: `jrly -install`

In this example, the default JRLY is installed as an NT Service and is displayed in the Service Control Manager (SCM) as **Jolt Relay**.

Example 2: `jrly -install MASTER`

In this case, an instance of JRLY is installed as an NT Service and is displayed in the SCM as **Jolt Relay_MASTER**. The suffix, MASTER, does not have any significance; it is only used to uniquely identify various instances of JRLYs.

Discussion: At this point, this instance of JRLY is not ready for starting. It must be assigned the configuration file (see the `set` command discussion) that specifies the listening TCP/IP port, JSH connection TCP/IP port, log files, and sockettimeout. This file should not be shared between various instances of JRLY.

`jrly -remove [display_suffix]  | -all`

Remove one or all JRLY from NT service.

If `[`*`display_suffix`*`]` is specified, this command removes the specified JRLY service.

If `[`*`display_suffix`*`]` is not specified, this command removes the default JRLY from being an NT Service.

If the `-all` option is specified, all the JRLY NT Services are removed. Related NT registry entries under:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
```

and

```
HKEY_LOCAL_MACHINE\Software\BEA\Jolt\1.2
```

are removed.

`jrly -set [-d `*`display_suffix`*`] -f `*`config_file`*

Update the registry with the full path of a new configuration file.

Example 1: `jrly -set -f c:\tux64\udataobj\jolt\jrly.con`

In this example, the default JRLY NT Service (Jolt Relay) is assigned a configuration file called `jrly.con` that is located in: `c:\tux64\udataobj\jolt` directory.

Example 2: `jrly -set -d MASTER -f c:\tux64\udataobj\jolt\master.con`

Here, the JRLY NT Service instance, called **Jolt Relay_MASTER** is assigned a configuration file called `jrly_master.con` that is located in `c:\tux64\udataobj\jolt` directory.

`jrly -manual [`*`display_suffix`*`]`

Set the start/stop to manual.

This command sets the specified JRLY instance to be manually controlled, using either the command-line options or the SCM.

`jrly -auto [`*`display_suffix`*`]`

Set the start/stop to automatic.

This command sets all the operations for specified NT Service to be automatically started when the OS boots and stopped when the OS shuts down.

`jrly -start [`*`display_suffix`*`]`

Start the JRLY.

This command starts the specified JRLY.

`jrly -stop [`*`display_suffix`*`]`

Stop the JRLY.

This command stops the specified JRLY.

`jrly -version`

Print the current version of JRLY.

This command prints the current version of JRLY binary.

`jrly -help`

Print command-line options.

This command prints the command-line options with brief descriptions.

JRLY Command-Line Option for UNIX

There is only one JRLY command-line option for UNIX:

```
jrly -f <config_file_path>
```

Start the JRLY process.

This option starts the JRLY process. If the configuration file does not exist or cannot be opened, the JRLY prints an error message. If the JRLY is unable to start, it writes a message to standard error, attempts to log the startup failure in the error log, then exits.

JRLY Configuration File

The format of the configuration file is a TAG=VALUE format. Blank lines or lines starting with a "#" are ignored. Refer to Listing 3-3 for an example of the formal specifications of the configuration file.

Listing 3-3 Specification of Configuration File

```
LOGDIR=<LOG_DIRECTORY_PATH>
ACCESS_LOG=<ACCESS_FILE_NAME in LOGDIR>
ERROR_LOG=<ERROR_FILE_NAME in LOGDIR>
LISTEN=<IP:Port combination where JRLY will accept connections>
CONNECT=<IP:Port combination associated with JRAD>
SOCKETTIMEOUT=<Seconds for socket accept()function>
```

Note: `SOCKETTIMEOUT` is the duration (in seconds) for which the relay NT service blocks for network activity (new connections, data to be read, closed connections). It is valid only on NT machines.

`SOCKETTIMEOUT` also affects the SCM. When the SCM requests the service to stop, the SCM needs to wait at least `SOCKETTIMEOUT` seconds before doing so.

Listing 3-4 shows an example of the JRLY configuration file. The CONNECT line specifies the IP address and port number of JRAD machine.

Listing 3-4 Example of JRLY Configuration File

```
LOGDIR=/usr/log/relay
ACCESS_LOG=access_log
ERROR_LOG=errorlog
# jrly will listen on port 4444
LISTEN=200.100.10.100:4444
CONNECT=200.100.20.200:4444, 200.100.20.200:5555,...
SOCKETTIMEOUT=30            //See note under Listing 3-4
```

The format for directory and file names is determined by the operating system. UNIX systems use the forward slash (/). NT systems use the backslash (\). If any of the files specified in `LOGDIR`, `ACCESS_LOG` or `ERROR_LOG` cannot be opened for writing, the JRLY prints an error message on `stderr` and exits.

The format for host names and port numbers are shown in Table 3-3.

Table 3-3 Host Name and Port Number Formats

Host Name/Port Number	Descriptions
`Hostname:Port`	`Hostname` is a string, `Port` is a decimal number
`//Hostname:Port`	`Hostname` is a string, `Port` is a decimal number
`IP:Port`	`IP` is a dotted notation IP address, `Port` is a decimal number

Jolt Relay Adapter

The Jolt Relay Adapter (back-end relay) is a TUXEDO system server. The Jolt Relay Adapter (JRAD) server may or may not be located on the same TUXEDO host machine (in SHM mode) and server group to which the JSL server is connected.

The JRAD can be started independently of its associated JRLY. JRAD tracks its startup and shutdown activity in the TUXEDO log file.

JRAD Configuration

A single JRAD process can only be connected to a single JRLY. A JRAD can be configured to communicate with only one JSL and its associated JSHs. However, multiple JRADs can be configured to communicate with one JSL. The `CLOPT` parameter for the TUXEDO servers must be included in the `UBBCONFIG` file. For additional information about the `CLOPT` parameters, refer to Table 3-4.

Table 3-4 JRAD CLOPT Parameter Descriptions

CLOPT Parameter	Description
`-l <hexadecimal format>`	Port to listen for the JRLY to connect on behalf of the client.
`-c <hexadecimal format>`	The address of the corresponding JSL to which JRAD connects.
`-H <hexadecimal format>`	Used when there is a network address translation performed for JRLY listen address.

Note: The format is `0x0002PPPPNNN`. Refer to the *BEA Jolt 1.2 Release Notes* for additional information on JRAD.

Listing 3-5 shows the sample `UBBCONFIG` file.

Listing 3-5 Sample JRAD Entry in UBBCONFIG File

```
# JRAD host 200.100.100.10 listens at port 2000, connects to JSL
port 8000 on the same host

JRAD    SRVGRP=JSLGRP   SRVID=60
        CLOPT="-A -- -l 0x000207D0C864640A –c 0x00021f40C864640A"
```

Network Address Configurations

There are several networked components that must be configured to work together when configuring a Jolt Internet Relay. Prior to configuration, review the criteria required in Table 3-5 and record the information. This will help minimize the possibility of misconfiguration.

Table 3-5 Jolt Internet Relay Network Address Configuration Criteria

JRLY	JRAD	JSL
LISTEN: <Location where the clients connect> CONNECT: <Location of your JRAD. Must match the `-l` parameter of JRAD>	`-l`: <Location of where the listener connects the JRLY> `-c`: <Location of JSL. Must match `-n` parameter of JSL>	`-n`: <Location of JSL. Must match `-c` parameter of JRAD>

Jolt Repository

The Jolt Repository contains TUXEDO service definitions that allow Jolt clients to access TUXEDO services. The Jolt Repository files included with the installation contain service definitions used internally by Jolt. See the *BEA Jolt Developer's Guide* for detailed instructions on how to add definitions to the application services.

Configuring the Jolt Repository

To configure the Jolt Repository, modify the application `UBBCONFIG` file. The `UBBCONFIG` file is an ASCII version of the TUXEDO configuration file. Create a new `UBBCONFIG` file for each application. See the *BEA TUXEDO Reference Manual* for information regarding the syntax of the entries for the file. Listing 3-6 shows relevant portions of the `UBBCONFIG` file.

Listing 3-6 Sample UBBCONFIG File

```
*GROUPS
JREPGRP        GRPNO=94 LMID=SITE1
*SERVERS
JREPSVR SRVGRP=JREPGRP SRVID=98
RESTART=Y GRACE=0 CLOPT="-A -- -W -P /app/jrepository"
JREPSVR SRVGRP=JREPGRP SRVID=97
RESTART=Y RQADDR=JREPQ GRACE=0 CLOPT="-A -- -P /app/jrepository"
JREPSVR SRVGRP=JREPGRP SRVID=96
RESTART=Y RQADDR=JREPQ REPLYQ=Y GRACE=0 CLOPT="-A -- -P
/app/jrepository"
```

Note: For UNIX systems, use the slash (`/`) when setting the path to the `jrepository` file (for example, app/repository). For NT systems, use the backslash (`\`) and specify the drive name (for example, `c:\app\repository`).

Change the sections of the UBBCONFIG file indicated in Table 3-6:

Table 3-6 UBBCONFIG File

Section	Parameters to be specified
GROUPS	LMID, GRPNO
SERVERS	SRVGRP, SRVID

GROUPS Section

A GROUPS entry is required for the group that includes the Jolt Repository. The group name parameter is a name selected by the application.

1. Specify the same identifiers given as the value of the LMID parameter in the MACHINES section.
2. Specify the value of the GRPNO between 1 and 30,000 in the GROUPS section.

SERVERS Section

The Jolt Repository server, JREPSVR, contains services for accessing and editing the Repository. Multiple JREPSVR instances share repository information through a shared file. Include JREPSVR in the SERVERS section of the UBBCONFIG file.

1. Indicate a new server identification (for example, 98) with the SRVID parameter.
2. Specify the -W flag for one JREPSVR to ensure that you can edit the Repository. The Repository is read-only without this flag.

   Note: You must install only one writable JREPSVR (that is, only one JREPSVR with the -W flag). Multiple read-only JREPSVRs can be installed on the same host.

3. Type the -P flag to specify the path of the repository file. An error message displays in the TUXEDO ULOG file if the argument for the -P flag is not entered.
4. Add the file pathname of the Repository file (for example, /app/jrepository).
5. Boot the TUXEDO system using the tmloadcf command (for example, tmloadcf -y ubbconfig) and tmboot command. See the *Administering the BEA TUXEDO System* for information on tmloadcf and tmboot.

Repository File

A Repository file, `jrepository`, is available with Jolt. This file includes `bankapp` services and the Repository services that you can modify, test, and delete using the Repository Editor.

Note: The Jolt 1.2 Repository file is different from the Jolt 1.1 Repository file. If you are using Jolt 1.1, you must make applicable changes.

Inside the `jrepository` file, the service definitions for the services in the Jolt Repository Server (JREPSVR) have FML32 as their buffer types. During installation, the new service definitions for Jolt 1.2 JREPSVR should be appended to the existing `jrepository` file as part of the upgrade.

Start with the `jrepository` file provided with the installation, even if you are not going to test the `bankapp` application with Jolt. Delete the `bankapp` packages or services that you do not need.

The pathname of the file must match the argument of the `-P` option.

STOP **Warning:** Do not modify the Repository files manually or you will not be able to use the Repository Editor. Although the `jrepository` file can be modified and read with any text editor, the Jolt system does not have integrity checks to ensure that the file is in the proper format. Any manual changes to the `jrepository` file might not be detected until runtime. See “Using the Jolt Repository Editor” in the *BEA Jolt Developer’s Guide* for additional information.

Initializing Services Using TUXEDO and the Repository Editor

You must initially define the TUXEDO services using TUXEDO and Jolt in order to make the Jolt services available to the client.

1. Build the TUXEDO server containing the service. See *Administering the BEA TUXEDO System* or *BEA TUXEDO Programmer’s Guide* for additional information on the following:

   - Building the TUXEDO applications/server

- Editing the `UBBCONFIG` file
- Updating the `TUXCONFIG` file
- Administering the `tmboot` command

2. Access the Jolt Repository Editor. See “Using the Jolt Repository Editor” in the *BEA Jolt Developer’s Guide* for additional information on the following:
    - Adding a Service
    - Saving Your Work
    - Testing a Service
    - Exporting/Unexporting Services

Event Subscription

Jolt Event Subscription is used to receive event notifications from either TUXEDO services or other TUXEDO clients:

Unsolicited Event Notifications. These are notifications that a Jolt client receives as a result of a TUXEDO client or service subscribing to unsolicited events, and a TUXEDO client issuing a broadcast (using either a `tpbroadcast()` or a directly targeted message via a `tpnotify()` ATMI call). Unsolicited event notifications do not need the `TMUSREVT` server.

Brokered Event Notifications. These notifications are received by a Jolt client via the TUXEDO Event Broker. The notifications are only received when both Jolt clients subscribe to an event and any TUXEDO client or server posts an event using `tppost()`. Brokered event notifications require the `TMUSREVT` server.

Configuring for Event Subscription

Configure the TUXEDO `TMUSREVT` server and modify the application `UBBCONFIG` file. Listing 3-7 shows the relevant sections of `TMUSREVT` parameters in the `UBBCONFIG` file. See the *BEA TUXEDO Programmer's Guide* for information regarding the syntax of the entries for the file.

Listing 3-7 UBBCONFIG File

```
TMUSREVT        SRVGRP=EVBGRP1  SRVID=40        GRACE=3600
                ENVFILE="/usr/tuxedo/bankapp/TMUSREVT.ENV"
                CLOPT="-e tmusrevt.out -o tmusrevt.out -A --
                -f /usr/tuxedo/bankapp/tmusrevt.dat"
                SEQUENCE=11
```

In the `SERVERS` section of the `UBBCONFIG` file, modify the `SRVGRP` and `SRVID` parameters as needed.

Filtering TUXEDO FML or VIEW Buffers

Filtering is a process that allows you to customize a subscription. If you require additional information about the TUXEDO Event Broker, subscribing to events, or filtering, refer to the *BEA TUXEDO Programmer's Guide, Volume 1*.

In order to filter TUXEDO FML or VIEW buffers, the field definition file must be available to TUXEDO at runtime.

Note: There are no special requirements for filtering STRING buffers.

Buffer Types

Table 3-7shows TUXEDO buffer types:

Table 3-7 TUXEDO Buffer Types

Buffer Type	Description
FML	Attribute, value pair. Explicit.
VIEW	C structure. Very precise offsetting. Implicit.
STRING	Length and offset are different values. All readable.
CARRAY	Character array. Blob of binary data. Only client and server know - JSL doesn't.
X_C_TYPE	Equivalent to VIEW.
X_COMMON	Equivalent to VIEW, but used for both COBOL and C.
X_OCTET	Equivalent to CARRAY.

FML Buffer Example

Listing 3-8 shows an example using the FML buffer. The FML field definition table is made available to TUXEDO by setting the `FIELDTBLS` and `FLDTBLDIR` variables.

To filter a field found in the `my.flds` file:

1. Copy the `my.flds` file to `/usr/me/bankapp` directory.

2. Add `my.flds` to the `FIELDTBLS` variable in the `TMUSREVT.ENV` file as shown in Listing 3-8:

Listing 3-8 FIELDTBLS Variable in the TMUSREVT.ENV File

```
FIELDTBLS=Usysflds,bank.flds,credit.flds,event.flds,my.flds
FLDTBLDIR=/usr/tuxedo/me/T6.2/udataobj:/usr/me/bankapp
```

If `ENVFILE="/usr/me/bankapp/TMUSREVT.ENV"` is included in the definition of the `UBBCONFIG` file (shown in Listing 3-7), the `FIELDTBLS` and `FLDTBLDIR` definitions are taken from the `TMUSREVT.ENV` file and not from your environment variable settings.

If you remove the `ENVFILE="/usr/me/bankapp/TMUSREVT.ENV"` definition, the `FIELDTBLS` and `FLDTBLDIR` definitions are taken from your environment variable settings. The `FIELDTBLS` and `FLDTBLDIR` definitions must be set to the appropriate value prior to booting the TUXEDO system.

For additional information on event subscriptions and the Jolt Class Library, refer to “Using the Jolt Class Library” in the *BEA Jolt Developer’s Guide*.

TUXEDO Background Information

The following sections provide detailed configuration information. Skip this section if you are familiar with BEA TUXEDO.

Configuration File

The TUXEDO configuration file for your application exists in two forms, the ASCII file, `UBBCONFIG`, and a compiled version called `TUXCONFIG`. Once you have created a `TUXCONFIG`, it is best to think of your `UBBCONFIG` as a backup.

You can make changes to the `UBBCONFIG` file with your preferred NT editor. Then, at a time when your application is not running, and when you are logged in to your MASTER machine, you can recompile your `TUXCONFIG` by running `tmloadcf`(1). System/T prompts you to make sure you really want to overwrite your existing `TUXCONFIG` file. (If you enter the command with the `-y` option, the prompt is suppressed.)

The UBBCONFIG File

A binary configuration file called the `TUXCONFIG` file contains information used by `tmboot(1)` to start the servers and initialize the bulletin board of a BEA TUXEDO system in an orderly sequence. The binary `TUXCONFIG` file cannot be created directly. Initially, you must create a `UBBCONFIG` file. That file is parsed and loaded into the `TUXCONFIG` using `tmloadcf`(1). Then `tmadmin`(1) uses the configuration file or a copy of it in its monitoring activity. `tmshutdown`(1) references the configuration file for information needed to shut down the application.

Configuration File Format

The `UBBCONFIG` file can consist of up to nine specification sections. Lines beginning with an asterisk (*) indicate the beginning of a specification section. Each such line contains the name of the section immediately following the *. Allowable section names are: `RESOURCES, MACHINES, GROUPS, NETGROUPS, NETWORK, SERVERS, SERVICES, INTERFACES, and ROUTING`.

Note: The RESOURCES (if used) and MACHINES sections *must* be the first two sections, in that order; the GROUPS section must be ahead of SERVERS, SERVICES, and ROUTING.

To configure the JSL, you must modify the UBBCONFIG file. For further information regarding TUXEDO configuration, refer to the *TUXEDO Administration Guide*. Listing 3-9 shows relevant portions of the UBBCONFIG file.

Listing 3-9 UBBCONFIG File

```
*MACHINES
MACH1  LMID=SITE1
       MAXWSCLIENTS=40
*GROUPS
JSLGRP        GRPNO=95   LMID=SITE1
*SERVERS
JSL SRVGRP=JSLGRP SRVID=30 CLOPT= “ -- -n 0x0002PPPPNNNNNNNN -d
/dev/tcp -m2 -M4 -x10”
```

The parameters shown in Table 3-8 are the only parameters that must be designated for the Jolt Server groups and Jolt Servers. You are not required to specify any other parameters.

Change the sections of the UBBCONFIG file shown in Table 3-8.

Table 3-8 UBBCONFIG File Sections

Section	Parameters to be specified
MACHINES	MAXWSCLIENTS
GROUPS	GRPNO, LMID
SERVERS	SRVGRP, SRVID, CLOPT

MACHINES Section

The `MACHINES` section specifies the logical names for physical machines for the configuration. It also specifies parameters specific to a given machine. The `MACHINES` section must contain an entry for each physical processor used by the application. Entries have the form:

`ADDRESS or NAME` *`required parameters [optional parameters]`*

where `ADDRESS` is the physical name of the processor, for example, the value produced by the UNIX system `uname -n` command.

`LMID=`*`string_value`*

This parameter specifies that the *`string_value`* is to be used in other sections as the symbolic name for `ADDRESS`. This name cannot contain a comma, and must be 30 characters or less. This parameter is required. There must be an `LMID` line for every machine used in a configuration.

`MAXWSCLIENTS=`*`number`*

The `MAXWSCLIENTS` parameter is required in the `MACHINES` section of the configuration file. It specifies the number of accesser entries on this processor to be reserved for Jolt and /WS clients only. The value of this parameter must be between 0 and 32,768, inclusive.

The Jolt Server and /WS use `MAXWSCLIENTS` in the same way. For example, if 200 slots are configured for `MAXWSCLIENTS`, this number configures TUXEDO for the total number of remote clients used by Jolt and /WS.

Be sure to specify `MAXWSCLIENTS` in the configuration file. If it is not specified, the default is 0.

Note: If `MAXWSCLIENTS` is not set, the JSL does not boot.

GROUPS Section

This section provides information about server groups, and must have at least one server group defined in it. A server group entry provides a logical name for a collection of servers and/or services on a machine. The logical name is used as the value of the `SRVGRP` parameter in the `SERVERS` section to identify a server as part of this group. `SRVGRP` is also used in the `SERVICES` section to identify a particular instance of a service with its occurrences in the group. Other GROUPS parameters associate this group with a specific resource manager instance (for example, the employee database). Lines within the GROUPS section have the form:

```
GROUPNAME required parameters [optional parameters]
```

where GROUPNAME specifies the logical name (string_value) of the group. The group name must be unique within all group names in the GROUPS section and LMID values in the MACHINES section. The group name cannot contain an asterisk(*), comma, or colon, and must be 30 characters or less.

A GROUPS entry is required for the group that includes the Jolt Server Listener (JSL). Make the GROUPS entry as follows:

1. The group name is selected by the application, for example: JSLGRP and JREPGRP
2. Specify the same identifiers given as the value of the LMID parameter in the MACHINES section.
3. Specify the value of the GRPNO between 1 and 30,000 in the *GROUPS section.

Note: Make sure that Resource Managers are *not* assigned as a default value for all groups in the GROUPS section of your UBBCONFIG file. Making Resource Managers the default value assigns a Resource Manager to the JSL and you receive an error during tmboot. In the SERVERS section, default values for RESTART, MAXGEN, etc., are acceptable defaults for the JSL.

SERVERS Section

This section provides information on the initial conditions for servers started in the system. The notion of a server as a process that continually runs and waits for a server group’s service requests to process may or may not apply to a particular remote environment. For many environments, the operating system, or perhaps a remote gateway, is the sole dispatcher of services. When either of these is the case, you need only specify SERVICE entry points for remote program entry points, and not SERVER table entries. BEA TUXEDO system gateway servers would advertise and queue remote domain service requests. Host-specific reference pages must indicate whether or not UBBCONFIG server table entries apply in their particular environments, and if so, the corresponding semantics. Lines within the SERVERS section have the form:

```
AOUT required parameters [optional parameters]
```

where AOUT specifies the file (string_value) to be executed by tmboot(1). tmboot executes AOUT on the machine specified for the server group to which the server belongs. tmboot searches for the AOUT file on its target machine, thus, AOUT must exist in a file system on that machine. (Of course, the path to AOUT can include RFS connections to file systems on other machines.) If a relative pathname for a server is given, the search for AOUT is done first in APPDIR, then in TUXDIR/bin, then in /bin, and then in <path>, where <path> is the value of the last PATH= line appearing in the machine environment file, if one exists. The values for APPDIR and TUXDIR are taken from the appropriate machine entry in the TUXCONFIG file.

Clients connect to Jolt applications through the Jolt Server Listener (JSL). Services are accessed through the Jolt Server Handler (JSH). The JSL supports multiple clients and acts as a single point of contact for all the clients to connect to the application at the network address that is specified on the JSL command line. The JSL schedules work for handler processes. A handler process acts as a substitute for clients on remote workstations within the administrative domain of the application. The handler uses a multiplexing scheme to support multiple clients on one port concurrently.

The network address specified for the JSL designates a TCP/IP address for both the JSL and any JSH processes associated with that JSL. The port number identified by the network address specifies the port number on which the JSL accepts new client connections. Each JSH associated with the JSL uses consecutive port numbers at the same TCP/IP address. For example, if the initial JSL port number is 8000 and there are a maximum of three JSH processes, the JSH processes use ports 8001, 8002, and 8003.

Note: Misconfiguration of the subsequent JSL results in a port number collision.

Parameters Usable With JSL

In addition to the parameters specified in the previous sections, the following parameters can be used with the JSL, although you need to understand how doing so would affect your application.

SVRGRP=*string_value*

This parameter specifies the group name for the group in which the server is to run. *string_value* must be the logical name associated with a server group in the *GROUPS section, and must be 30 characters or less. This association with an entry in the *GROUPS section means that AOUT is executed on the machine with the LMID specified for the server group. This association also specifies the GRPNO for the server group and parameters to pass when the associated resource manager is opened. All server entries must have a server group parameter specified.

SRVID=*number*

This parameter specifies an *identifier*, an integer between 1 and 30,00, inclusive, that identifies this server within its group. This parameter is required on every server entry, even if the group has only one server. If multiple occurrences of servers are desired, do not use consecutive numbers for SRVIDs; leave enough room for the system to assign additional SRVIDs up to MAX.

Optional Parameters

The optional parameters of the SERVERS section are divided into boot parameters and runtime parameters.

Boot Parameters

Boot parameters are used by tmboot when it executes a server. Once running, a server reads its entry from the configuration file to determine its runtime options. The unique server identification number is used to find the right entry. The following are boot parameters.

CLOPT=*string_value*

The CLOPT parameter specifies a string of command line options to be passed to AOUT when booted.The servopts(5) page in the *TUXEDO Reference Manual:* Section 5 lists the valid parameters.

Some of the available options apply primarily to servers under development. For example, the -r option directs the server to write a record to its standard error file each time a service request begins or ends.

Other command line options may be used to direct to server’s standard out and standard error to specific files, or to start the server so that it initially advertises a limited set of its available services.

The default value for the CLOPT parameter is -A, which means that the server is started with all available services advertised.

The maximum length of the `CLOPT` parameter value is 256 characters; it must be enclosed in double quotes.

`SEQUENCE=`*`number`*

This parameter specifies when this server should be booted or shutdown relative to other servers. If `SEQUENCE` is not specified, servers are booted in the order found in the `SERVERS` section (and shut down in the reverse order). If some servers have sequence numbers specified and others do not, all servers with sequence numbers are booted first from low to high sequence number, then all servers without sequence numbers are booted in the order in which they appear in the configuration file. Sequence numbers must be in the range between 1 and 9999. If the same sequence number is assigned to more than one server, `tmboot` may boot those servers in parallel.

`MIN=`*`number`*

The `MIN` parameter specifies the minimum number of occurrences of the server to boot by `tmboot`. If an `RQADDR` is specified, and `MIN` is greater than 1, the servers form a Multiple Servers Single Queue (MSSQ) set. The identifiers for the servers are `SRVID` up to (`SRVID` + (`MAX` -1)). All occurrences of the server have the same sequence numbers as well as any other server parameters. The value range for `MIN` is 0 to 1000. If `MIN` is not specified, the default value is 1.

`MAX=`*`number`*

The `MAX` parameter sets the maximum number of occurrences of the server to be booted. Initially, `tmboot` boots `MIN` servers, and additional servers can be booted up to `MAX` occurrences using the `-i` option of `tmboot` to specify the associated server identifier. The value range for `MAX` is 0 to 1000. If no value is specified for `MAX`, the default is the same as for `MIN`, or 1. Keep in mind that:

- `tmboot` starts `MIN` occurrences unless you explicitly call for more with the `-i SRVID` option of `tmboot`
- If `RQADDR` is specified and `MIN` is greater than one, an MSSQ set is formed
- If `MIN` is not specified, the default is 1
- If `MAX` is not specified, the default is `MIN`
- `MAX` is especially important for conversational servers because they are spawned automatically as needed

Runtime Parameters

The runtime parameters are used by the server after it has been started by `tmboot`. As indicated above, `tmboot` uses the values found in the `TUXDIR`, `APPDIR` and `ENVFILE` parameters for the `MACHINES` section when booting the server. It also sets the `PATH` for the server to:

```
        “APPDIR:TUXDIR/bin:/bin:<path>”
```

where `<path>` is the value of the last `PATH=` line appearing in the `ENVFILE` file. The following parameters are runtime parameters.

ENVFILE=*string_value*

The `ENVFILE` parameter for a server can be used to add values to the environment established by `tmboot` during initialization of the server. Variables specified in the file named in the `SERVERS ENVFILE` parameter are set after those in the `MACHINES ENVFILE` used by `tmboot`. These files cannot be used to override `TUXDIR, APDIR`, `TUXCONFIG`, or `TUSOFFSET`. The best policy is to include in the server’s `ENVFILE` only those variable assignments known to be needed to ensure proper running of the application.

Note that on the server, this file is processed *after* the server starts. Therefore, it cannot be used to set the pathnames used to find executable or dynamically loaded files needed to execute the server. If you need to perform these tasks, use the machine `ENVFILE` instead.

Within `ENVFILE` only lines of the form

        `VARIABLE =`*string*

are allowed. `VARIABLE` must start with an underscore or alphabetic character and can contain only underscore or alphanumeric characters. If the server is associated with a server group that can be migrated to a second machine, the `ENVFILE` must be in the same location on both machines.

CONV={Y | N}

`CONV` specifies whether or not the server is a conversational server. `CONV` takes a `Y` value if a conversational server is being defined. Connections can only be made to conversational servers, and rpc requests (via `tpacall(3c)` or `tpcall(3c)`) can only be made to non-conversational servers. For a request/response server, you can either set `CONV=N`, which is the default, or omit the parameter.

RQADDR=*string_value*

RQADDR assigns a symbolic name to the request queue of this server. MSSQ sets are established by using the same symbolic name for more than one server (or by specifying MIN greater than 1). All members of an MSSQ set must offer an identical set of services and must be in the same server group.

If RQADDR is not specified, the system assigns a unique key to serve as the queue address for this server. However, tmadmin commands that take a queue address as an argument are easier to use if queues are given symbolic names.

RQPERM=*number*

The RQPERM parameter is used to assign UNIX-style permissions to the request queue for this server. The value of *number* can be between 0001 and 0777, inclusive. If no parameter is specified, the permissions value of the bulletin board, as specified by PERM in the RESOURCES section, is used. If no value is specified there, the default of 0666 is used (this opens your application up to possible use by any login on the system, so consider this carefully).

REPLYQ={ Y | N }

The REPLYQ parameter specifies (with a Y or N) whether or not a reply queue, separate from the request queue, should be established for AOUT. If Y is specified, the reply queue is created on the same LMID as the AOUT. In cases where only one server is using the request queue, replies can be picked up from the request queue without causing problems. However, if the server is a member of an MSSQ set and contains services programmed to receive reply messages, REPLYQ should be set to Y so that an individual reply queue is created for this server. If that is not done, the reply will be sent to the request queue shared by all servers for the MSSQ set and there is no way of ensuring that it will be picked up by the server that is waiting for it.

It should be standard practice for all member servers of an MSSQ set to specify REPLYQ=Y if replies are anticipated. Servers in an MSSQ set are required to have identical offerings of services, so it is reasonable to expect that if one server in the set expects replies, any server in the set can also expect replies.

RPPERM=*number*

The RPPERM parameter is used to assign permissions to the reply queue. *number* is specified in the usual UNIX fashion (for example, 0600); the value can be between 0001 and 0777, inclusive. If RPPERM is not specified, the default value 0666 is used. This parameter is useful only when REPLYQ=Y. If requests and replies are read from the same queue, only RQPERM is needed; RPPERM is ignored.

RESTART={ Y | N }

The RESTART parameter takes a Y or N to indicate whether or not AOUT is restartable. The default is N. If the server is in a group that can be migrated, RESTART must be Y. Note that a server started with a SIGTERM signal cannot be restarted; it must be rebooted.

An application's policy on restarting servers might vary according to whether the server is in production or not. During the test phase of application development it is reasonable to expect that a server might fail repeatedly, but server failures should be rare events once the application has been put into production. You might want to set more stringent parameters for restarting servers once the application is in production.

Parameters Associated With RESTART

RCMD=*string_value*

If AOUT is restartable, this parameter specifies the command that should be executed when AOUT abnormally terminates. The string, up to the first space or tab, must be the name of an executable UNIX file, either a full pathname or relative to APPDIR. (Don't attempt to set a shell variable at the beginning of the command.) The command name may be optionally followed by command-line arguments. Two additional arguments are appended to the command line: the GRPNO and SRVID associated with the restarting server. *string_value* is executed in parallel with restarting the server.

The RCMD parameter can be used to specify a command to be executed in parallel with the restarting of the server. The command must be an executable UNIX system file residing in a directory on the server's PATH. An example of a possible use would be a command that sends a customized message to the userlog to mark the restarting of the server.

MAXGEN=*number*

If AOUT is restartable, this parameter specifies that it can be restarted at most (number - 1) times within the period specified by GRACE. The value must be greater than 0 and less than 256. If not specified, the default is 1 (which means that the server can be started once, but not restarted). If the server is to be restartable, MAXGEN must be equal to or greater than 2. RESTART must be Y or MAXGEN is ignored.

GRACE=*number*

If RESTART is Y, the GRACE parameter specifies the time period (in seconds) during which this server can be restarted, (MAXGEN - 1) times. The number assigned must be equal to or greater than 0, and less than 2,147,483,648 seconds (or a little more than 68 years). If GRACE is not specified the default is 86,400 seconds (24 hours). Setting GRACE to 0 removes all limitations; the server can be restarted an unlimited number of times.

Entering Parameters

TUXEDO parameters, including RESTART, RQADDR, and REPLYQ, can be used with the JSL. (See *Administering the BEA TUXEDO System* for additional information regarding runtime parameters.) Enter the following parameters:

1. To identify the SRVGRP parameter, type the previously defined group name value from the GROUPS section.

2. To indicate the SRVID, type a number between 1 and 30,000 that identifies the server within its group.

3. Verify that the syntax for the CLOPT parameter is as follows:

   ```
   CLOPT= “-- -n 0x0002PPPPNNNNNNNN -d /dev/tcp -m2 -M4 -x10”
   ```

   Note: The CLOPT parameters may vary. Refer to Table 3-2 for pertinent command-line information.

4. If necessary, type the optional parameters:

   - Type the SEQUENCE parameter to determine the order that the servers are booted.
   - Specify Y to permit release of the RESTART parameter.
   - Type 0 to permit an infinite number of server restarts using the GRACE parameter.

Sample Applications in Jolt Online Resources

You can access sample code that can be modified for use with Jolt through the BEA Jolt product Web page at:

`http://www.beasys.com/products/jolt/index.htm`

These samples demonstrate and utilize Jolt features and functionality.

Other Web sites with Java-related information include:

- Javasoft Home Page (`http://www.java.sun.com/`)
- In addition, the newsgroups in the comp.lang.java hierarchy contain lists of past articles and communications regarding Java, and are a valuable source of archival material.

CHAPTER

4 Post Installation

This chapter discusses post installation procedures, and includes the following sections:

- Installing JRLY After Normal Installation
- Uninstalling Jolt

Installing JRLY After Normal Installation

The following sections give instructions for installing JRLY.

Installing JRLY on UNIX

JRLY is a stand-alone process provided to run on the same machine as the Web server. On UNIX systems, no changes have been made, and JRLY functions as it did in previous versions of Jolt.

Installing JRLY on NT

On NT systems, when you install JRLY as an NT Service, BEA Jolt 1.2 registers the configuration file when NT boots.

On NT, if you do not install JRLY with the normal Jolt installation, but decide you want it later, you can install it through configuration on the command line with the following command:

1.
```
jrly -install [display_suffix]    //registers JRLY
```

2.
```
jrly -set [-d display_suffix] -f <config file>     //registers
                                                //configuration file
```

3.
```
jrly -manual [display_suffix]      //sets start/stop to manual
```

   or

```
jrly -auto [display_suffix]       //Sets the start/stop
                                  // to automatic
```

Uninstalling Jolt

The following sections describe how to uninstall Jolt.

Uninstalling Jolt From NT

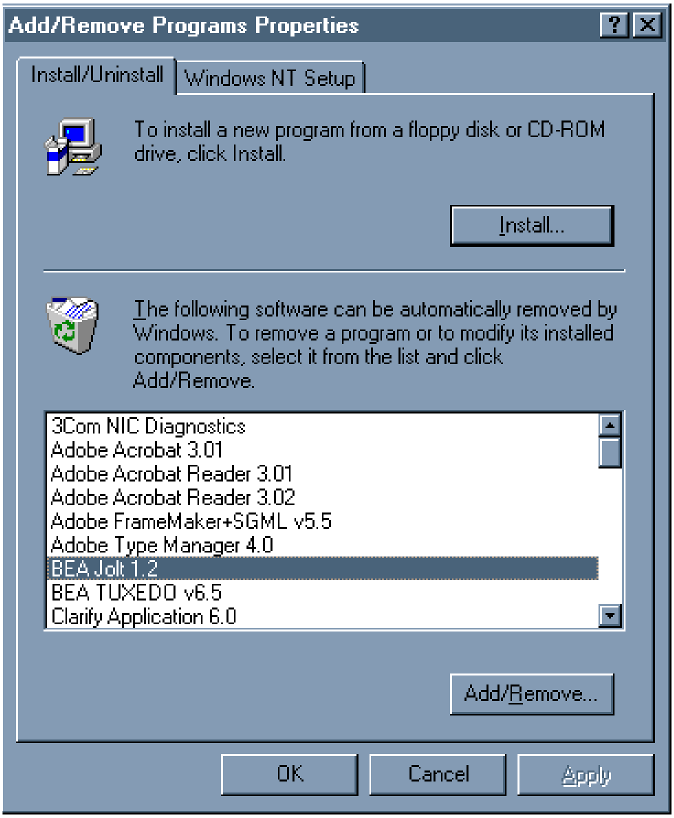
To remove Jolt, follow these instructions:

1. From your **Control Panel**, select **Add/Remove Programs**.

   The **Add/Remove Programs Properties** dialog box appears (See Figure 4-1).

2. From the list of programs shown, select “BEA Jolt”.

Figure 4-1 The Add/Remove Programs Properties Dialog Box



To install a new program from a floppy disk or CD-ROM drive, click Install.

The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

3Com NIC Diagnostics
Adobe Acrobat 3.01
Adobe Acrobat Reader 3.01
Adobe Acrobat Reader 3.02
Adobe FrameMaker+SGML v5.5
Adobe Type Manager 4.0
BEA Jolt 1.2
BEA TUXEDO v6.5
Clarify Application 6.0

3. Click on the **Add/Remove** button.

   A confirmation box appears and asks if you really want to remove the selected application.

4. The **Remove Programs from Your Computer** dialog box appears and the program is removed.

   When the program has been removed, the following message appears in the dialog box: “Uninstall Successfully Completed.”

5. Click the **OK** button.

6. Click the **OK** button in the **Add/Remove Programs Properties** dialog box.

   Jolt has been removed.

Uninstalling Jolt From UNIX

Currently, there is no script for uninstalling Jolt from UNIX. You can, however, uninstall Jolt from your UNIX system by removing the applicable files and directories.

Index