BEA Tuxedo®

Release Notes

Release 8.1
Part Number: 801-001002-009
Revised: April 2007

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

BEA Tuxedo 8.1 Release Notes

BEA Tuxedo Release 8.1
Date: December 2007

Table 1 Revision History

Revision Date	Summary of Change
December 2007	Added the following certifications (see BEA Tuxedo 8.1 Platform Data Sheets): ■ HP-UX 11iv3 (B.11.31) (32-bit) Itanium ■ HP-UX 11iv3 (B.11.31) (32-bit) PA-RISC
October 2007	Added the following certification (see BEA Tuxedo 8.1 Platform Data Sheets): ■ HP-UX 11iv3 (B.11.31) (64-bit) Itanium
June 2007	Marked the following platforms as EOL by Operating System Provider (see BEA Tuxedo 8.1 Platform Data Sheets): ■ Compaq Tru64 UNIX Version 5.1a on Alpha Systems: EOL by Operating System Provider ■ Compaq Tru64 UNIX Version 5.1b (64-bit) on Alpha Systems: EOL by Operating System Provider ■ Compaq Tru64 UNIX Version 5.1b-1 (64-bit) on Alpha Systems: EOL by Operating System Provider ■ Compaq Tru64 UNIX Version 5.1b-2 (64-bit) on Alpha Systems: EOL by Operating System Provider ■ HP-UX Version 11.0 (32-bit) on HP 9000 Series: EOL by Operating System Provider IBM AIX 4.3.3

Table 1 Revision History

Revision Date	Summary of Change
April 2007	Added the following platforms (see BEA Tuxedo 8.1 Platform Data Sheets): ■ HP-UX 11iv1 (B.11.11) (64-bit) on PA-RISC
January 2007	Added the following to IBM AIX 5.3 (32/64-bit) on 64-bit IBM PowerPC (see BEA Tuxedo 8.1 Platform Data Sheets): ■ Note: The `-qnamemangling=v5` flag is required for compiling CORBA applications using Visual Age C++ 6.0 Added the following to IBM AIX 5.3 (64-bit) on IBM PowerPC ■ Note: The `-qnamemangling=v5` flag is required for compiling CORBA applications using IBM(R) XL C/C++ Enterprise Edition V7.0
October 2006	Added the following Certified Platforms (see BEA Tuxedo 8.1 Platform Data Sheets): ■ Compaq Tru64 UNIX Version 5.1b (64-bit) on Alpha Systems ■ Compaq Tru64 UNIX Version 5.1b-1 (64-bit) on Alpha Systems ■ Compaq Tru64 UNIX Version 5.1b-2 (64-bit) on Alpha Systems ■ HP-UX 11iv2 (B.11.23) (32/64-bit) on PA-RISC ■ HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC ■ IBM AIX 5.3 (32/64-bit) on 64-bit IBM PowerPC ■ IBM AIX 5.3 (64-bit) on IBM PowerPC ■ Red Hat Enterprise Linux 4.0 (32-bit) on Pentium ■ Solaris 9 (32/64-bit) on SPARC ■ Solaris 9 (64-bit) on SPARC ■ Solaris 10 (32/64-bit) on SPARC ■ Solaris 10 (64-bit) on SPARC ■ Turbo Linux DS 10 (32-bit) on Pentium
August 2006	Marked the following platforms as EOL by Operating System Provider (see BEA Tuxedo 8.1 Platform Data Sheets): ■ IBM AIX 4.3.3 ■ Microsoft Windows 98 (client) ■ Red Hat Linux 7.2

Table 1 Revision History

Revision Date	Summary of Change
December 2005	■ Added CR238486, CR242065, and CR244891 to “Product Constraints” as Constraint number 13, 14, and 15 respectively. ■ Added CR236565, CR241622, CR244880, and CR244894 to “Known Problems in BEA Tuxedo 8.1.” ■ Updated Appendix A, BEA Tuxedo 8.1 Platform Data Sheets, of Installing the BEA Tuxedo System. The revised document can be found at the following URL: http://edocs.bea.com/tuxedo/tux81/install/inspds.htm ■ Updated “Software Supported by Each Platform” in this document for post-release platform data information. Added following post-release platforms (see BEA Tuxedo 8.1 Platform Data Sheets): • HP-UX 11iv1 (B11.11) (32/64-bit) on PA-RISC • HP-UX 11iv1 (B.11.11)(32-bit) Using -AA Option on PA-RISC • HP-UX 11iv2 (B.11.23) (64-bit) on Itanium • IBM AIX 5.2 (64-bit) on IBM PowerPC • Microsoft Windows 2003 Server (32-bit) on Pentium • Microsoft Windows 2003 Server (64-bit) on Itanium • Red Hat Linux Advanced Server 2.1 (32-bit) on Pentium • Red Hat Linux Advanced Server 2.1 (64-bit) on Itanium • Red Hat Linux Advanced Server 3.0 (32-bit) on Pentium • Red Hat Linux Advanced Server 3.0 (64-Bit) on Itanium 2 • Solaris 8 (64-bit) on SPARC • Solaris 10 (32-bit) on x86 (64-bit AMD64) • SUSE Linux Enterprise Server 8.0 (32-bit) on IBM zSeries • SUSE Linux Enterprise Server 9.0 (32-bit) on Pentium

Table 1 Revision History

Revision Date	Summary of Change
September 2005	■ Added CR236565, CR238601, and CR241622 to "Known Problems in BEA Tuxedo 8.1." ■ Added CR240505 to "Product Constraints" as Constraint number 12.
August 2005	■ Added CR221229 and CR236719 to "Product Constraints" as Constraint number 10 and 11.
July 2004	■ Updated link to platform support information in "Supported Platforms."
September 2003	■ Updated compiler information in "Software Supported by Each Platform."
June 2003	■ Added CR108651 to "Documentation Addenda."
May 2003	■ Added enhancement to MBSTRING feature in "MBSTRING Enhancement."
April 2003	■ Added new section "Hot Upgrade From Tuxedo 8.0 to Tuxedo 8.1." ■ Added limitations to MBSTRING feature in "Multibyte Character Support Limitations."
January 2003	Initial Release

This document contains release notes for the BEA Tuxedo 8.1 release, including ATMI, CORBA, BEA Jolt, and SNMP Agent.

Release Notes Topics

This document includes the following topics:

- About This BEA Tuxedo Release
- Software and Documentation Problems Fixed in This Release

- BEA Tuxedo Software Components
- Software Component Licensing Requirements
- Supported Platforms
- Deprecated Features in BEA Tuxedo 8.1
- Software Environment
- Online Documentation
- Product Constraints
- Known Problems in BEA Tuxedo 8.1
- Documentation Addenda
- Online Help Addenda
- How to Obtain Patches

About This BEA Tuxedo Release

BEA Tuxedo software provides businesses and organizations that depend on mission-critical applications with the flexibility of two proven programming interfaces: an Application-to-Transaction Monitor Interface (ATMI) and a Common Object Request Broker Architecture (CORBA) interface. Both interfaces use the BEA Tuxedo infrastructure, which has demonstrated, through years of use in large, transaction-based, production systems, that it is powerful, robust, scalable, manageable, and reliable.

This topic includes the following sections:

- What's New and Improved
- Installation Upgrade Considerations
- Unsupported Code Samples and Tools Web Page

What's New and Improved

BEA Tuxedo Release 8.1 includes the following new features and enhancements:

- Localized install for Japan

  BEA Tuxedo is now localized for the Japanese environment through installation screens and message catalogs.

- Multibyte character encoding

  A new buffer type and FML field type known as MBSTRING allows BEA Tuxedo APIs to send strings that consist of multibyte characters. In addition, BEA Tuxedo passes the character set encoding name and MBSTRINGs are automatically convert from one encoding representation to another.

  Note: Refer to "Multibyte Character Support Limitations," for limitations to the MBSTRING feature.

- XML parser integration

  BEA Tuxedo includes the public Xerces XML parser on the software DR-ROM. The Xerces XML parser is enhanced to include local DTD cache.

- Single point security administration option

  BEA Tuxedo and BEA WebLogic Tuxedo Connector (WTC) can share a common user database which is stored in a BEA WebLogic Server LDAP database.

- Domain gateway performance improvement

  The Tuxedo Domain gateway is streamlined to provide up to 20% faster throughput.

  Note: Specific performance improvements are highly application-dependent.

- Remote Domain connection policy enhancement

  Each remote domain can have its own `CONNECTION_POLICY` setting.

- Domains Keepalive feature

  You can enable TCP and Application-level keepalive for a domain gateway connection to provide better response to network or system failures.

- Multi Threaded Bridge for improved performance using the BRTHREADS parameter

  You can set the Bridge to run as a multithreaded process allowing it to use more than one CPU on a multiprocessor system. This can result in increased throughput of up to 10%.

  Note: Specific performance improvements are highly application-dependent.

- Parameter length enhancement

  The maximum length of parameters in the UBBCONFIG and DMCONFIG files representing most pathnames or network addresses have increased to 256 bytes.

- Global maximum transaction timeout parameter

  A new parameter, `MAXTRANTIME`, is available to limit the transaction timeout value system-wide so transactions will not have unreasonably lengthy timeouts.

- CORBA C++ client enhancement

  The C++ client now supports WebLogic Server load balancing and method level failover. WebLogic Server clusters can provide IIOP clients with information about what nodes in the cluster support a given object. That information along with a load balancing algorithm selected in the deployment of a WebLogic application, allow the C++ client to select a node in the cluster to process a request. That information is also used to retry a request in the event of a communication or system failure. This last feature is called method level failover, as the client will try the request against the remaining node(s) in the cluster that has the object available to it.

For a discussion of each these features, see the What’s New link on the BEA Tuxedo online documentation.

Installation Upgrade Considerations

Before installing the product, be sure to review the product issues in “Known Problems in BEA Tuxedo 8.1” on page 49. For complete information on upgrading to BEA Tuxedo 8.1, see “Upgrading the BEA Tuxedo System to Release 8.1” in *Installing the BEA Tuxedo System*.

Hot Upgrade From Tuxedo 8.0 to Tuxedo 8.1

In order to perform a hot upgrade from BEA Tuxedo 8.0 to BEA Tuxedo 8.1, you must be running Tuxedo 8.0 rolling patch 22 at a minimum.

Unsupported Code Samples and Tools Web Page

BEA Tuxedo customers can download unsupported code samples and developer tools from the BEA Systems, Inc. dev2dev online site. Start on the following Web page:

`http://dev2dev.bea.com/index.jsp`

If you do not already have a BEA dev2dev login, links are provided with instructions for free membership.

Software and Documentation Problems Fixed in This Release

The following tables list the software and documentation problems that have been fixed in Release 8.1. Problems are listed by CR (Change Request) number.

Any software or documentation fixes that are made to Release 8.1 after these *BEA Tuxedo Release 8.1 Release Notes* are printed will be reported in updates to this section in the online version of these *Release Notes*. The online version is available through the BEA Tuxedo 8.1 Release Notes link at

`http://e-docs.bea.com`

Problems Fixed in This Release

Table 2 lists problems fixed in BEA Tuxedo 8.1.

Table 2 BEA Tuxedo Problems Fixed in This Release

CR Number	Former Problem Description
CR074806	Memory leak in JSH.
CR084000	Semaphore lock held erroneously during reboot.
CR010923	'-L' option of tuxwsvr does not work.
CR019890	tmipcrm don't work on HP.
CR020645	NT/WS client ULOG loops with errors when timeout with NOTIFY=SIGNAL.
CR021278	Service Info class missing from package.bea.jolt.pool.
CR031578	endsession() in Jolt client receives SessionException when tpnotify from server.
CR032113	Jolt client hangs when no handlers are available.
CR033430	Internet Relay new session takes much longer on NT that in jolt1.1.
CR035387	Potential deadlock on AIX platform due to code optimization.
CR043625	Service request dropped from non-Master node to another non-Master.
CR045294	MARSHALL exception in Java with objref in struct in an Any.
CR045312	GWTDOMAIN LIBGW_CAT:5301: NLS:6: Message - TPEPROTO.
CR046319	ISL fails to boot from backup in MP mode when authentication is used.
CR046582	Memory corruption when using security with domains.
CR047603	Application Servers leak memory.
CR048180	ISL won't time out all ocm entries if > 5000 table entries.
CR048378	JSH core dumps with signal 11 after tpcancel tperrno TPEBADDESC.
CR048410	ATMI call can hang in multithreaded, single-context client.
CR048527	GWTDOMAIN may core dump when connection retried after long network outage.
CR048564	Prevent infinite ISL loop for asymmetric outbound IIOP (OCM table).
CR048738	restartsrv may cause application server to shutdown.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR049480	Services added via DM_MIB do not show up in tmadmin.
CR049489	java.lang.NullPointerException from service call
CR049514	Add 'Multithreaded' option in BuildTuxedo tool (NT) for Server builds.
CR049579	Per-thread memory leak in a multi-threaded client.
CR049946	Tuxedo 8.0 server fails to boot with Tux 6.5 if compression limit set.
CR050406	Generated valuetype _i files have double deletion in copy_value().
CR050492	Handle leak in JSH when service returns.
CR050516	typecodes have incompatible type for value factories.
CR050593	Any insertion for WStringValue not defined.
CR050664	Cannot unmarshall sequence of Any's.
CR050678	WStringValue's not marshalled under GIOP 1.0.
CR050760	RMI-IIOP Null values not recognized correctly.
CR050841	idl compiler core dumps on Tru64 for nested valuetype idl.
CR050901	"_ec_sec_atz_authorize_preop" function can't be reallocated.
CR051024	Any insertion for ValueBase not defined.
CR051253	GWTDOMAIN may crash on restart.
CR051319	A thread may hang in tpalloc with multi-threaded native client.
CR051434	Interoperability problem between Tux 8 and WLE 5.1.
CR051456	Tuxedo 7.1 fails to connect Tuxedo 8.0 domain with APP_PW security.
CR051507	Tuxedo does not generate or detect byte-order markers in wide strings
CR051519	Tuxedo 8.0 server may send mixed endian messages to WLS.
CR051530	UTF-16 not encoded correctly.
CR051694	Tuxedo 8.0 Master Node throws LIBTUX:CAT:6185 when the NonMaster node is Tux6.4.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR051698	Tuxedo 6.4 native client unable to get Broadcast message from Tuxedo 8.0 server.
CR051717	GWTDOMAIN under load, unexpected error messages.
CR051780	tppost for Tux6.5 to Tux8.0 with VIEW buffer fails with LIBTUX_CAT:406 error.
CR051832	GWTDOMAIN can go in a tight loop.
CR051833	xa_recover() not called when doing a tmshutdown -k KILL TMSGRP.
CR051847	Workstation client memory leak in tpinit-tpterm loop.
CR051919	Service timeout changes dynamically when there is MIB activity.
CR051946	Propagate the changes in Tux6.5 for CR048738 and CR051850 for restartsrv.
CR051949	WS Client: TPSIGRSTRT flag does not work.
CR052050	Window 2000 Event log messages.
CR052063	Tuxedo CNS won't work in context of a transaction.
CR052131	Recursive tpacall across domains may hang the application.
CR052201	GWTDOMAIN memory leaks in connection protocol.
CR052362	Error storm:LIBTUX_CAT:489.
CR052555	TMS_SQL fails to boot in AIX.
CR052653	JSH crashes when client requests a service name longer than 15 chars.
CR052875	SSL tests can't load SSL library on AIX.
CR053041	Service Name different on AIX.
CR053072	"Duplicate Symbol" Warning in buildserver in AIX load01.
CR053118	GWTDOMAIN fails to advertise remote services.
CR053154	use TypeCode's equal vs. Java's byte wise equals to compare typecodes.
CR053296	Jolt repository editor locks up when duplicate parameters exists.
CR053299	Union member of object on OBV gets wrong discriminant.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR053300	Marshal/Unmarshal gets exception for custom valuetype.
CR053301	Accessing large array on ObjectByValue shows problem.
CR053367	JSH fails to boot with JOLT_CAT:1079 error in AIX 4.3.3.
CR053456	_e_sec_atn_gss_accept_sec_context not get called by self auth.
CR053920	WSC hang at getmsg.
CR053975	Fboolev() may not evaluate certain expressions correctly.
CR054179	dmloadcf/tmboot does not handle mixed remote domain configurations.
CR054392	shutdown/restartsrv problems, including CMDTUX_CAT:944 and LIBTUX_CAT:577.
CR054562	On HP platforms, dumpmem creates 0 byte files and corrupt file names >=24.
CR054659	Tuxedo processes consume large amount of CPU on Compaq Alpha GS Series (NUMA) machines.
CR054746	LIBTUX 6311 if SEC_PRINCIPAL_NAME length is greater than 30 characters.
CR055013	ISH crash when doing a "tmadmin broadcast" command.
CR055026	Shutdown of domains may hang sometimes under load.
CR055029	Recovery of Transaction blob returns length but buffer is filled with NULL.
CR055085	Few libgw functions always return -1 back to caller.
CR055178	IIOP Client hang forever if server shutdown.
CR055781	tpgetrply() on Tuxedo 8.0 fails with compression on and nonMaster is tux 6.5.
CR055799	cns command-line tools hang on AIX
CR055800	University samples need updates for AIX.
CR056875	tmshutdown -w5 -y fails when queue/blocked server encountered.
CR057064	WSH crashes on Solaris 2.7.
CR057430	Unicode escapes not accepted in typeids.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR057476	No _ptr definition generated for some valuetypes.
CR057480	valuetype inheriting from an abstract valuetype doesn't generate OBV.
CR057554	tmadmin config edit loses CLOPT -t parameter.
CR057568	C++ client with no SSL, doesn't work against WLS IOR's with SSLSecTransComponent.
CR057793	BBMAXTKT increased by 13 times.
CR058256	GWTDOMAIN may not retry connection to remote domains in some situations.
CR058299	Use `->id()` method on typecode instead of hard coded reposid in _register method.
CR058405	FML32 request times out during interop testing TUX7.1<->TUX65.
CR058519	servers stuck in the SHUTDOWN RESTARTING/CLEANING or DEAD state.
CR059049	TMUSREVT does not distribute event message to subscriber.
CR059654	socket leaks per JoltSession when used with jrly/JRAD.
CR059682	SDK license works with buildserver but not with buildtuxedo.
CR059731	IIS ASP session pool Result class missing methods.
CR059749	DBBL migration does not work.
CR059777	USER_AUTH security and concurrency per request : CORBA/INTERNAL.
CR060059	Very long symbol names caused IDL compiler to crash.
CR060563	unchecked state of BuildTuxedo 'Multithreaded' check box not properly saved/restored.
CR060881	GWTDOMAIN memory leak when using DDR.
CR060881	Memory leak in domains when using DDR (different from CR060881).
CR061057	WSH registry slot cleanup.
CR061208	JRLY-JRAD communication performance degradation.
CR061215	WLE 5.1 CORBA Domain can't access TUX 8.0 CORBA Domain.
CR061324	Link errors when using USDEFERLEVEL macro in ATMI client.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR061416	tpcall cores when SYSTEM_ACCESS is set to PROTECTED
CR061735	LIBWSC 1034+1032 when ubb(NOTIFY=SIGNAL) on tpcall.
CR061897	Fappend32 does not work with FLD_FML32.
CR061909	tppost fails with TPESVCFAIL and LIBTUX_CAT:1553: in ULOG.
CR061911	Repeated tuxgetenv calls slow down performance.
CR061960	GWTDOMAIN may fail to failover to alternative network addr.
CR062074	jrly does not handle signals.
CR062116	jrly does not reconnect to JRAD after a network outage.
CR062187	"adv -d ldom service" fails after multiple remote domains.
CR062473	Typedef of longlong causes exception in OBB_create_tc() at initialization.
CR062491	Stale messages don't get drained in multithreaded server.
CR062726	Multithreaded WS client cores when calling _tmfmsg_free.
CR062818	Administration Console (webgui) hangs while loading applet.
CR062851	when querying local M.I.B, some fields are not displayed.
CR062864	Conversation may not get disconnected in certain situations.
CR062880	tpcall dumps core when ACL is turned on.
CR062975	ISL -K with USER_AUTH gives ISNAT_CAT:1289.
CR063027	TMQFORWARD -d not working.
CR063920	Memory leak in WSH, when tpinit() fails.
CR063962	Global transactions time out when MAXTRAN is exceeded.
CR064151	multi-threaded /WS client crash at tpacall after the WSL -T idle timeout.
CR064154	CORBA Java client got ArrayIndexOutOfBoundsException in marshalling.
CR064290	The attribute of TA_SERVERCTXT class is not returned correctly.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR064317	If server not booted at connection time, reconnection does not work.
CR064744	JRLY performance degradation still exists under heavy load.
CR065323	Second TP get reply, hangs after previous one works.
CR065447	returns handle 0 from tpacall.
CR065780	Incorrect GWTDOMAIN tperrno.
CR066234	TMUSREVT problems when many subscribers die between sanity.
CR066497	Problem with TMSYSEVT in writing system userlog messages.
CR066659	server-to-client (outbound) compression CMPLIMIT/TMCMPPRFM is not working.
CR067393	tpnotify(with TPACK) call during tpcall execution causes hangs.
CR068335	TMQFORWARD drop messages with certain kernel MSGMAX and MSGMNB settings.
CR068567	Unexpected NullPointerException thrown by Jolt (duplicate of CR049489).
CR06862	memory-leak at CORBA::COMM_FAILURE exception delivery/re-throw.
CR069002	WLEC securitycontext parameters not propagated to WLE 5.1 /Tux 8.0.
CR069340	GWTDOMAIN Tux 6.5 interop problem when encoding on for 64bit machines.
CR069495	The hard coded limit of 1000 service names in txrpt.
CR069497	/WS Incorrect tpurcode received by the client from AUTHSVR using USER_AUTH (1).
CR070118	BBL hangs up on Windows.
CR070330	Fboolev fail with CARRAY whose length is in multiple of 4.
CR070485	When querying local MIB, some fields are not displayed.
CR070505	Finer TMQFORWARD idletime resolution (< 1 second).
CR070737	tmloadcf and tmshutdown don't recognize y/n responses.
CR071066	Userlog() does not recover from an open() /write() failure.
CR071271	tmadmin pnw displays incorrect information on slave node.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR071310	Stale message clean up, when svctimeout invoked - msg remains in queue
CR071458	Remote domain fails to disconnect.
CR071499	Network send error during network retry.
CR072126	The attribute of TA_TIMELEFT class is not returned correctly.
CR072325	C++ idl problem with _default() in Boolean Unions.
CR072566	tmadmin hung forever when DBBL/Bridge is congested due to network outage.
CR073398	Appkey in Tuxedo 7.1 and 8.0 are incorrect after installation of RP50 for NT/2000.
CR073697	WS client hang forever during network outage.
CR073751	Memory leak when using TPMULTICONTEXTS.
CR074103	Preventive fix for AOM table corruption (CR064706).
CR074508	Message presend fails with multi-threaded/multi-context WSC to Tux 6.5 server.
CR074802	Remote nodes stuck after "LIBTUX_CAT:1276: WARN: Forcing deletion of table entries for server ..."
CR074820	/WS Incorrect tpurcode received by the client from AUTHSVR using USER_AUTH (2).
CR074827	server in TMS group crashes running in PROTECTED mode without an explicit tpopen() in the tpsrvinit().
CR075014	/WS client generates LIBWSC_CAT:1526 error when signal is caught and WSL -N is specified.
CR075037	Cannot dynamically add/update the eLink OSI TP domain's NWADDR field via dmadmin.
CR075223	Jolt shutdown class does not work.
CR075611	WSC memory leaks when Multi-threaded client used.
CR075964	tpchkauth() generates error for multiple tpinit()/tpterm() from same client
CR076026	GWTDOMAIN intermittently core dumps.
CR076135	LIBGWT_CAT: 1563 reported instead of 1553.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR076136	LIBGWT_CAT: 1560 reported instead of 1554.
CR076294	WSL crashes if the -H address passed has a certain length.
CR076580	Data mismatch between clients.
CR076774	handle leak when using Fldid32() or Fname32().
CR076872	Windows 2000 : slisten cannot start when service table is full.
CR077073	GWADM exiting upon reception of SIGALRM.
CR077118	GWTDOMAIN re-connect failed after DM_PW was updated via DM_MIB.
CR077570	sbbl and BBL race condition.
CR077796	TMS cannot be shutdown when a timeout occurs and second tmshutdown may hang forever.
CR077970	Message LIBTUX_CAT:4055 with unreadable service name may appear in certain situations.
CR078440	fix compiler warning messages for digital unix platform.
CR078550	BB deadlock (when network partitioned and app servers killed).
CR078623	LIBWSC_CAT:1355 generated when X_C_TYPE buffer is used in tpcall.
CR078729	JSH uses up CPU and runs slow with large FML32 buffers (850K).
CR078845	dmadmin adv command --> wrong associated function.
CR078908	Cannot dynamically add/update the eLink OSI TP domain's XATMI_ENCODING field via dmadmin.
CR079041	GWTDOMAIN core dump.
CR079225	Domain Data Dependent Routing fails for FML buffer type.
CR080330	JSH incorrectly logs a debug message.
CR080361	JRAD dies after first client connect.
CR080379	tpcall with FML data buffer causes LIBTUX_CAT:406: ERROR: Message decode failure.
CR080420	CORBA server cores with SYSTEM_ACCESS set to PROTECTED.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR081104	Wrong version of AUTHSRV.c included with NT 4.0/W2K base installation.
CR081448	CORBA C++ #pragma version and forward reference problem.
CR081546	TA_NUMTRAN of T_SERVER class can not be retrieved with multi threaded.
CR081627	"Wk Queued" not cleaned up when svctimeout invoked.
CR081958	GWTDOMAIN reconnect failed.
CR082063	txrpt output incorrect in Tuxedo 6.5 64-bit on AIX 4.3.3.
CR082100	tptypes() returns TPEINVAL in _tminitbuf for user-defined buffer types.
CR082150	Bridge core dump with big message buffers when CMPLIMIT compression on.
CR082183	If security is USER_AUTH, tpinit may always fail after unsuccessful login attempt.
CR082303	ULOGPFX does not redirect LIBTUX_CAT:6126 messages.
CR082429	Bridge process in Windows 2000 gives GPF while doing CLEaning operation after Non Master is partitioned.
CR082429	Bridge crashes in MP configuration if pclean is done on failed secondary.
CR082536	Server loops unexpectedly under specific tmboot/tmshutdown sequence
CR082648	/Q space becomes insane if shutdown TMQFORWARD while in transaction.
CR082740	JSH memory leak.
CR082895	WTC tpcall fails randomly while Tuxedo service completes successfully.
CR082927	Default appkey different from documentation in multi-domain.
CR083182	Unsol msgs cause tpterm() failures, tpinit() threads not cleaned up.
CR083194	tpinit/tpterm process losing VM memory when done in a loop.
CR083217	tpacall without tpgetrply yields invalid service increment.
CR083403	FML32 buffer many times bigger than expected.
CR083554	Fappend32 does not work with FLD_VIEW32.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR083607	Multi-threaded server dumps core due to incorrect message from GWTDOMAIN on remote disconnect.
CR084035	Factory can't be found after secondary node crashes
CR084063	HP aCC compilers complain about "future errors".
CR084064	tpacall with flag TPNOREPLY yields invalid service increment.
CR084184	Memory Leak with TPMULTICONTEXTS and FML.
CR084259	Fcmp32 pretends 2 different carray fields are the same.
CR084296	THREAD-notification client into loop when disconnecting the network.
CR084340	DDR continues to wildcard /TDOMAIN even after configured /TDOMAIN becomes unavailable.
CR084346	security credential propagation at SecurityLevel1 may fail with NO_PERMISSION.
CR084750	C++ IDL compiler crashes on a struct containing a union.
CR084827	ud32/wud32 fails to find field name.
CR085055	tpacalls from WSC can get LIBWSC_CAT:1019, wrong tperrno.
CR085373	Cannot re-attach to the bulletin board when a large BBL is configured.
CR085592	JSH writes JOLT_CAT:1133 under heavy load and when encryption and security turned on.
CR085791	tpcall consumes 100% CPU when network is down.
CR085801	tmshutdown -w causes LIBTUX:4055:ERROR.
CR086118	tpcall hangs up on WSL shutdown when tpinit flag is TPU_THREAD.
CR086207	tmadmin/pq gives wrong info. when cleanupsrv being called.
CR086554	TA_CURREQ returns negative values.
CR086572	Premature time-outs.
CR087146	tpgetrply may return TPEBLOCK instead of TPESYSTEM when WSH is down.
CR087158	Memory leak using DDR with XML.

Table 2 BEA Tuxedo Problems Fixed in This Release (Continued)

CR Number	Former Problem Description
CR087626	IDL compiler fails to compile some IDL generated by WLS 6.1 EJBs IDL.
CR088561	BBL may miss SVCTIMEOUT.
CR088896	threads concurrency: Per-context mutex not released during blocking calls.
CR089326	GWTDOMAIN memory leak.

MBSTRING Enhancement

The following `MBSTRING` enhancement (CR103546) is available in BEA Tuxedo 8.1 rolling patch 19. This enhancement enables `MBSTRING` to be self-describing if `sendlen` is set to zero.

Some Tuxedo buffers provide a capability for the buffer to determine its own length if the user does not provide it. This self-describing behavior is triggered when an application sets the `sendlen` argument of a Tuxedo function call (for example, `tpcall()`) to zero. Rolling Patch 19 adds this self-describing capability for `MBSTRING.`

Note: Refer to "How to Obtain Patches," for information on how to obtain rolling patches.

Implementation

This self-describing behavior is implemented by adding the following:

- `_mbspresend()` function to the `MBSTRING` typeswitch function list
- A protective file, `$TUXDIR/udataobj/sendlen0_unsafe_tpmbenc`, containing the names of the codeset encoding names which are considered unsafe to use with the feature.

The `_mbspresend()` addition requires any user who customizes Tuxedo buffers to rebuild their applications.

Safe/Unsafe Encoding Names

The idea of safe or unsafe encoding names specified by `TPMBENC` comes from whether or not the multibyte character data for these encodings can contain embedded NULLs. Because the `_mbspresend()` function uses `strlen()` to determine the length of the data, an embedded NULL causes the length to be incorrectly set and the wrong number of data bytes are sent.

The default list in `sendlen0_unsafe_tpmbenc` has the multibyte Unicode encoding names (in uppercase and lowercase, for convenience) which can contain embedded NULLs. You should modify this list as application administration or performance is considered.

- If the file exists but is empty, then Tuxedo assumes all encoding names are safe for `MBSTRING` self-describing attempts.
- If the file does not exist, then Tuxedo assumes that all encoding names are unsafe to use for `MBSTRING` self-describing attempts (`tperrno` will be `TPEINVAL`).
- If the file exists and contains a list of names, then Tuxedo reads this file once (at `_mbsinit()`) and stores the list internally. During `mbsinit()`, the `TPMBENC` name is compared to the stored list and the buffer is set as safe or unsafe to use. When `_mbspresend()` is called (`sendlen` argument is set to zero) and the buffer is marked safe, then the length is set internally by Tuxedo.

BEA Tuxedo Software Components

The BEA Tuxedo software consists of the following components:

- BEA Tuxedo ATMI software

  The ATMI software enables you to build scalable ATMI applications using either of two programming languages: C or COBOL. This software includes the following components:

  - BEA Tuxedo ATMI servers
  - BEA Tuxedo /WS clients
  - BEA Tuxedo Native clients

    - BEA Tuxedo infrastructure

- BEA Tuxedo CORBA software

  The CORBA software enables you to build scalable CORBA applications in the C++ programming language. This software includes the following components:

    - CORBA C++ servers
    - C++ client and server Object Request Broker (ORB)
    - BEA Tuxedo object infrastructure

- BEA Tuxedo Administration Console

- BEA Jolt 8.1 software

  BEA Jolt is a Java-based interface to the BEA Tuxedo system that extends the functionality of existing BEA Tuxedo applications to include intranet- and Internet-wide availability.

- BEA SNMP Agent software

  BEA SNMP Agent for BEA Tuxedo and BEA WebLogic Enterprise is a Simple Network Management Protocol (SNMP) agent that enables BEA Tuxedo and BEA WebLogic Enterprise applications to be managed from an Enterprise Management Console.

- BEA Tuxedo 56-bit or 128-bit Encryption Package software that provides Secure Sockets Layer (SSL) and Link-Level Encryption (LLE) for BEA Tuxedo applications. This software is included in the BEA Tuxedo 8.1 distribution and is enabled or disabled depending on which license is used.

Software Component Licensing Requirements

For BEA Tuxedo 8.1, all software components are included on the product CD-ROMs. A single license is issued when you purchase the product that enables the components that you want to use.

Licensing is used to enable the product components as follows:

- Basic license

  This license enables the following components:

  - The ATMI and CORBA programming environments (clients and servers)
  - Secure Sockets Layer (SSL), Link-Level-Encryption (LLE), and Public Key Interface (PKI) plug-ins
  - 56-bit encryption

- 128-bit encryption license

  This license enables 128-bit encryption as well as the components enabled by the basic license.

Note: It will not be possible for a customer to use 128-bit encryption for data messages without obtaining a 128-bit encryption license from BEA. However, 128-bit encryption can be used for BEA Tuxedo Administration Console messages without obtaining a 128-bit license.

- Jolt license

  This license enables the BEA Jolt software as well as the components enabled by the basic license.

- Full license

  This license enables all product components.

BEA Tuxedo 8.1 requires that all customers upgrade their licenses to a new format; previous licenses will not work. Customers with existing support contracts may use one of the following methods to obtain a license upgrade:

- If you have a previous version of a BEA Tuxedo license, you may update your license from the BEA Support site using an automatic form, or by contacting a BEA representative.
- If you have a BEA WebLogic Enterprise license, you must contact your BEA representative to upgrade your license to BEA Tuxedo 8.1.

Supported Platforms

BEA Tuxedo software runs on the platforms listed in the following sections. BEA has certified these platforms for development and production use with the BEA Tuxedo release 8.1 product. BEA can provide customer support only for these platforms. Note that although BEA has attempted to implement the BEA Tuxedo software in a manner that conforms to industry-standards, it is not feasible for BEA to certify its use with all third-party databases, ORBs, and other products.

Additional software ports and certifications may continue after the initial release of BEA Tuxedo 8.1. For information regarding subsequent ports and certifications, please refer to the Platform Support information on the BEA web site at the following link:

`http://www.bea.com/products/tuxedo/platforms.shtml`

Platform information is maintained under the “Requirements” option under the Tuxedo product page.

Note: More detailed platform information is maintained on the secured eSupport portal under “Product News and EOL Updates.” A customer eSupport password login is required. The eSupport link is:

`http://support.bea.com`

BEA Tuxedo Server Platforms

The BEA Tuxedo server components run on the following platforms:

- Microsoft Windows 2000 Advanced Server (32-bit) on Pentium

  Note: Microsoft Windows Advanced Server (32-bit) data sheet information is applicable to Microsoft Windows XP (Professional) systems.

- Solaris 8 (32-bit) on SPARC
- HP-UX 11iv1 (B.11.11)(32-bit) Using -AA Option on PA-RISC
- HP-UX 11iv1 (B11.11) (32/64-bit) on PA-RISC

- HP-UX 11iv1 (B.11.11) (64-bit) on PA-RISC
- HP-UX 11iv2 (B.11.23) (32-bit) on Itanium
- HP-UX 11iv2 (B.11.23) (64-bit) on Itanium
- HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC
- HP-UX 11iv2 (B.11.23) (32/64-bit) on PA-RISC
- HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC
- HP-UX 11iv3 (B.11.31) (32-bit) Itanium
- HP-UX 11iv3 (B.11.31) (64-bit) Itanium
- HP-UX 11iv3 (B.11.31) (32-bit) PA-RISC
- IBM AIX 5.2 (32/64-bit) on IBM PowerPC
- IBM AIX 5.2 (64-bit) on IBM PowerPC
- IBM AIX 5.3 (32/64-bit) on 64-bit IBM PowerPC
- IBM AIX 5.3 (64-bit) on IBM PowerPC
- Microsoft Windows 2003 Server (32-bit) on Pentium
- Microsoft Windows 2003 Server (64-bit) on Itanium
- Red Flag Linux 4.1 (32-bit) on Pentium
- Red Hat Linux Advanced Server 2.1 (32-bit) on Pentium
- Red Hat Linux Advanced Server 2.1 (64-bit) on Itanium
- Red Hat Linux Advanced Server 3.0 (32-bit) on Pentium
- Red Hat Linux Advanced Server 3.0 (64-Bit) on Itanium 2
- Red Hat Enterprise Linux 4.0 (32-bit) on Pentium
- Solaris 8 (64-bit) on SPARC
- Solaris 9 (32/64-bit) on SPARC
- Solaris 9 (64-bit) on SPARC
- Solaris 10 (32-bit) on x86 (64-bit AMD64)

- Solaris 10 (32/64-bit) on SPARC
- Solaris 10 (64-bit) on SPARC
- SUSE Linux Enterprise Server 8.0 (32-bit) on IBM zSeries
- SUSE Linux Enterprise Server 9.0 (32-bit) on Pentium
- Turbo Linux DS 10 (32-bit) on Pentium

BEA Tuxedo Client Platforms

The BEA Tuxedo client software runs on the following platforms:

- BEA Tuxedo Native CORBA C++ clients:
  All server platforms listed in the previous section.
- BEA Tuxedo remote (IIOP) CORBA C++ clients:
  All server platforms listed in the previous section, plus Microsoft Windows 98.
- BEA Tuxedo /WS clients:
  All server platforms, plus Microsoft Windows 98 (EOL by Operating System Provider).
- BEA Tuxedo Native clients:
  All server platforms listed in the previous section.

BEA Tuxedo Administration Console Platforms

The BEA Tuxedo Administration Console software runs on all the platforms listed in the section “BEA Tuxedo Server Platforms” on page 24.

Note: While the Administration Console software cannot be installed on Microsoft Windows 98 systems, you can use the Web browser on your Microsoft Windows 98 system to access and use the Administration Console software on any BEA Tuxedo server system that is accessible over your network.

BEA Security Service Platforms

The BEA Tuxedo Security Service (56-bit or 128-bit) runs on the following platforms:

- Microsoft Windows 2000 Advanced Server (32-bit) on Pentium

  Note: Microsoft Windows Advanced Server (32-bit) data sheet information is applicable to Microsoft Windows XP (Professional) systems.

- Solaris 8 (32-bit) on SPARC
- HP-UX 11iv1 (B.11.11)(32-bit) Using -AA Option on PA-RISC
- HP-UX 11iv1 (B11.11) (32/64-bit) on PA-RISC
- HP-UX 11iv1 (B.11.11) (64-bit) on PA-RISC
- HP-UX 11iv2 (B.11.23) (32-bit) on Itanium
- HP-UX 11iv2 (B.11.23) (64-bit) on Itanium
- HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC
- HP-UX 11iv2 (B.11.23) (32/64-bit) on PA-RISC
- HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC
- HP-UX 11iv3 (B.11.31) (32-bit) Itanium
- HP-UX 11iv3 (B.11.31) (64-bit) Itanium
- HP-UX 11iv3 (B.11.31) (32-bit) PA-RISC
- IBM AIX 5.2 (32/64-bit) on IBM PowerPC
- IBM AIX 5.2 (64-bit) on IBM PowerPC
- IBM AIX 5.3 (32/64-bit) on 64-bit IBM PowerPC
- IBM AIX 5.3 (64-bit) on IBM PowerPC
- Microsoft Windows 2003 Server (32-bit) on Pentium
- Microsoft Windows 2003 Server (64-bit) on Itanium
- Red Flag Linux 4.1 (32-bit) on Pentium

- Red Hat Linux Advanced Server 2.1 (32-bit) on Pentium
- Red Hat Linux Advanced Server 2.1 (64-bit) on Itanium
- Red Hat Linux Advanced Server 3.0 (32-bit) on Pentium
- Red Hat Linux Advanced Server 3.0 (64-Bit) on Itanium 2
- Red Hat Enterprise Linux 4.0 (32-bit) on Pentium
- Solaris 8 (64-bit) on SPARC
- Solaris 9 (32/64-bit) on SPARC
- Solaris 9 (64-bit) on SPARC
- Solaris 10 (32-bit) on x86 (64-bit AMD64)
- Solaris 10 (32/64-bit) on SPARC
- Solaris 10 (64-bit) on SPARC
- SUSE Linux Enterprise Server 8.0 (32-bit) on IBM zSeries
- SUSE Linux Enterprise Server 9.0 (32-bit) on Pentium
- Turbo Linux DS 10 (32-bit) on Pentium

BEA Tuxedo 8.1 offers two type encryption services: secure sockets layer (SSL) and link-level encryption (LLE). The SSL and LLE encryption software is included on the BEA Tuxedo 8.1 distribution and is an integral part of the installation procedure. You also configure the SSL software during the installation.

Before you can use BEA Tuxedo Security Service software on any of the platforms listed above, you must first install it according to one of the following options:

- Option 1: On the Tuxedo server platforms, install at least one of the following BEA Tuxedo 8.1 server components:
  - Full Install
  - Server Install
- Option 2: On all platforms, install at least one of the following BEA Tuxedo 8.1 client components:
  - Client Install (All BEA Tuxedo client components; this is recommended)

- Customized Install
  - CORBA C++ client
  - BEA Tuxedo /WS client
  - BEA Jolt client

Software Environment

The following sections list the software that can run on each platform supported by the BEA Tuxedo software.

Software Supported by Each Platform

Table 3 lists the software supported by each platform that can run the BEA Tuxedo software

Table 3 Software Supported by Each Platform

Platform	Java 2 SDK and JRE[a]	C/C++ and COBOL Compilers	Clients	Servers
HP-UX 11iv1 (B.11.11)(32-bit) Using -AA Option on PA-RISC	Java 2 JRE 1.3.1; Java 2 SDK 1.3.1	aCC:HP aC++/ANSI C B3910B A.05.53 [Oct. 2 2003] cc: HP aC++/ANSI C B3910B A.05.50 [May 15 2003]	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv1 (B11.11) (32/64-bit) on PA-RISC	Java 2 JRE 1.3.1.08; Java 2 SDK 1.3.1.08	cc B.11.11.08 aCC A.03.50 with Patch (PHSS_28880 PHSS_28871)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI

Table 3 Software Supported by Each Platform (Continued)

Platform	Java 2 SDK and JRE[a]	C/C++ and COBOL Compilers	Clients	Servers
HP-UX 11iv1 (B.11.11) (64-bit) on PA-RISC	Java 2 JRE 1.3.0; Java 2 SDK 1.3.0	aCC: HP ANSI C++ B3910B A.03.13 cc: HP ANSI C B3910B A.11.01.07	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv2 (B.11.23) (64-bit) on Itanium	Java 2 JRE 1.3.1; Java 2 SDK 1.3.1	aCC: HP aC++/ANSI C B3910B A.05.53 [Oct 2 2003]	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv2 (B.11.23) (64-bit) on Itanium	Java 2 JRE 1.3.1; Java 2 SDK 1.3.1	aCC: HP aC++/ANSI C B3910B A.05.53 [Oct 2 2003]	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC	Java 2 JRE 1.3.1; Java 2 SDK 1.3.1	aCC: HP ANSI C++ B3910B A.03.13	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv2 (B.11.23) (32/64-bit) on PA-RISC	Java 2 JRE 1.4.2; Java 2 SDK 1.4.2	aCC (HP ANSI C++ B3910B A.03.60 cc (HP92453-01 B.11.11.12 HP C Compiler)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv2 (B.11.23) (64-bit) on PA-RISC	Java 2 JRE 1.4.2; Java 2 SDK 1.4.2	aCC (HP ANSI C++ B3910B A.03.60 cc (HP92453-01 B.11.11.12 HP C Compiler)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv3 (B.11.31) (32-bit) Itanium	Java 2 JRE 1.5.x; Java 2 SDK 1.5.x	cc: HP C/aC++ B3910B A.06.12 [Nov 03 2006]	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /W	CORBA C++; ATMI

Table 3 Software Supported by Each Platform (Continued)

Platform	Java 2 SDK and JRE[a]	C/C++ and COBOL Compilers	Clients	Servers
HP-UX 11iv3 (B.11.31) (64-bit) Itanium	Java 2 JRE 1.5.x; Java 2 SDK 1.5.x	cc: HP C/aC++ B3910B A.06.12 [Nov 03 2006]	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
HP-UX 11iv3 (B.11.31) (32-bit) Itanium	Java 2 JRE 1.5.x; Java 2 SDK 1.5.x	cc: B.11.59.01 061205 aCC: HP ANSI C++ B3910B A.03.33 COBOL: Micro Focus 5.0, ACUCOBOL-GT 7.0.1, or other compatible COBOL compiler.]	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
IBM AIX 5.2 (32/64-bit) on IBM PowerPC	Java full version "J2RE 1.3.1 IBM AIX build ca131-2002070"	VisualAge C++ Professional / C for AIX Compiler, Version 5.0.2 or later.	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
IBM AIX 5.2 (64-bit) on IBM PowerPC	Java 2 JRE 1.3.1; Java 2 SDK 1.3.1	VisualAge C/C++ for AIX Compiler, Version 5.0.2 or later.	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
IBM AIX 5.3 (32/64-bit) on 64-bit IBM PowerPC	Java 2 JRE 1.3.1 Java 2 SDK 1.3.1	Visual Age C/C++ 6.0.0.0 Note: The `-qnamemangling=v5` flag is required for compiling CORBA applications using Visual Age C++ 6.0	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI

Table 3 Software Supported by Each Platform (Continued)

Platform	Java 2 SDK and JRE[a]	C/C++ and COBOL Compilers	Clients	Servers
IBM AIX 5.3 (64-bit) on IBM PowerPC	2 JRE 1.4.1 (JDK 1.3.1 for CORBA java client) 2 SDK 1.4.1 (JDK 1.3.1 for CORBA java client)	C/C++: IBM(R) XL C/C++ Enterprise Edition V7.0 Note: The `-qnamemangling=v5` flag is required for compiling CORBA applications using IBM(R) XL C/C++ Enterprise Edition V7.0	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Microsoft Windows 2000 Advanced Server (32-bit) on Pentium* *Microsoft Windows Advanced Server (32-bit) data sheet information is applicable to Microsoft Windows XP (Professional) systems.	Java 2 SDK 1.3.0-C; Java 2 JRE 1.3.0-C; Java 2 SDK 1.3 IDL ORB (run time, non-BEA CORBA Java clients)	Microsoft Visual C++ 6.0 with Service Pack 4 or later; Microsoft Visual Basic 6.0[b] with Service Pack 4 or later; Net Express 3.1 (COBOL) from Merant or other compatible COBOL compiler	C++, ActiveX, CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS; ActiveX	CORBA C++; ATMI
Solaris 8 (32-bit) on SPARC	Java 2 JRE 1.3.0-C; Java 2 SDK 1.3 IDL ORB (run time, non-BEA CORBA Java clients)	Microsoft Visual C++ 6.0 with Service Pack 4 or later; Microsoft Visual Basic 6.0 with Service Pack 4 or later; Net Express 3.1 (COBOL) from Merant or other compatible COBOL compiler	CORBA Java over IIOP; CORBA C++ over IIOP; Tuxedo /WS; ActiveX	None

Table 3 Software Supported by Each Platform (Continued)

Platform	Java 2 SDK and JRE[a]	C/C++ and COBOL Compilers	Clients	Servers
Microsoft Windows 2003 Server (32-bit) on Pentium	Java 2 JRE 1.4.2; Java 2 SDK 1.4.2	Microsoft (R) 32-bit C/C++ Standard Compiler Version 13.10.3077 for 80x86	C++, ActiveX, CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS; ActiveX	CORBA C++; ATMI
Microsoft Windows 2003 Server (64-bit) on Itanium	Java 2 JRE 1.4.2; Java 2 SDK 1.4.2	Microsoft (R) C/C++ Optimizing Compiler Version 13.10.2240.8 for IA-64	C++, ActiveX, CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS; ActiveX	CORBA C++; ATMI
Solaris 8 (32-bit) on SPARC	Java 2 SDK 1.3.0; Java 2 JRE 1.3.0; Java 2 SDK 1.3 IDL ORB (run time, non-BEA CORBA Java clients)	C++ gcc version egcs-2.96.66 19990314/Linux (egcc-1.1.2 release) or later; Server Express 1.0.0 (COBOL) from Merant or other compatible COBOL compiler (required for development only)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Red Hat Linux Advanced Server 2.1 (32-bit) on Pentium	Java 2 JRE 1.4.1; Java 2 SDK 1.4.1	gcc/g++ version 2.96 20000731	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Red Hat Linux Advanced Server 2.1 (64-bit) on Itanium	Java 2 JRE 1.4.1; Java 2 SDK 1.4.1	gcc version 2.96 20000731 (Red Hat Linux 7.2 2.96-112.7.2)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Red Hat Linux Advanced Server 3.0 (32-bit) on Pentium	Java 2 JRE 1.4.1; Java 2 SDK 1.4.1	gcc/g++ version 3.2.3 20030502 (Red Hat Linux 3.2.3-24)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI

Table 3 Software Supported by Each Platform (Continued)

Platform	Java 2 SDK and JRE[a]	C/C++ and COBOL Compilers	Clients	Servers
Red Hat Linux Advanced Server 3.0 (64-Bit) on Itanium 2	Java 2 JRE 1.4.1; Java 2 SDK 1.4.1	gcc version 3.2.3 20030502 (Red Hat Linux 3.2.3-24)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Solaris 8 (32-bit) on SPARC	Java 2 SDK 1.3.0; Java 2 JRE 1.3.0; Java 2 SDK 1.3 IDL ORB (run time, non-BEA CORBA Java clients)	Sun WorkShop 6 C++ 5.1 (Forte 6.0 Update #2) or later; Server Express 1.0.0 (COBOL) from Merant or other compatible COBOL compiler (required for development only)	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Solaris 8 (64-bit) on SPARC	Java 2 JRE 1.3.0; Java 2 SDK 1.3.0	Sun WorkShop 6 update 2 C/C++ 5.3 2001/05/15	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
Solaris 10 (32-bit) on x86 (64-bit AMD64)	Java 2 JRE 1.5.0_01; Java 2 SDK 1.5.0_01	Sun Studio 10. Must apply patch: 117831, 117846, 118682	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
SUSE Linux Enterprise Server 8.0 (32-bit) on IBM zSeries	Java 2 JRE 1.4.1; Java 2 SDK 1.4.1	gcc/g++ 3.2.2	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI
SUSE Linux Enterprise Server 9.0 (32-bit) on Pentium	Java 2 JRE 1.4.2; Java 2 SDK 1.4.2	gcc 3.3.3	CORBA Java over IIOP; CORBA C++ over IIOP; CORBA C++ Native; Tuxedo /WS	CORBA C++; ATMI

[a] The Java 2 JRE is needed for the run-time environment.

[b] This is required only for client systems that run ActiveX client applications. It is not required for C++, Java, and VisiJava client systems or server-only Microsoft Windows 2000 systems.

Database Support

BEA Tuxedo ATMI and CORBA C++ applications support the XA standard. This facilitates inter-operation with any XA-compliant software system including database management systems.

Security Related Software Supported

The following security software is supported on all BEA Tuxedo platforms:

- SSL Certificate authorities
  - Verisign
  - Netscape
- To support certificate-based authentication when using SSL, BEA Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server that is included with the Netscape Enterprise Server

Table 4 lists the BEA Tuxedo clients and connections that support SSL when the SSL Certificate software is installed and the clients and connections that are not supported.

Table 4 BEA Tuxedo Support for SSL 3.0

SSL is supported for...	SSL is not supported for...
BEA Tuxedo CORBA C++ and IIOP clients.	BEA Tuxedo ATMI /WS client connections to the BEA Tuxedo 8.1 Workstation Listener/Handler (WSL/WSH).[a] BEA Jolt client connections to BEA Jolt Listener/Handler (JSL/JSH).

[a] BEA Tuxedo8.1 56-bit or 128-bit encryption is available for link-level encryption of these connections and also for link-level encryption of connections between machines and domains.

BEA Tuxedo End-of-Life Information

BEA Systems, Inc. periodically finds it necessary to discontinue support for certain older products to ensure the highest level of quality and support for our customers going forward. BEA has a policy of providing advanced notification to our customers so migration strategies and plans can be made.

To access end-of-life (EOL) information for the BEA Tuxedo product, access the BEA eSupport web site at the following link.

`http://support.bea.com`

Log in to the Support site or register to get a login ID to access EOL information. After logging in, click the Product News and EOL Information link in the left navigation area of the Support page.

Deprecated Features in BEA Tuxedo 8.1

When a product feature is deprecated, it is identified as a feature that will not be supported and may be removed in the next release of the product.

The following features are being deprecated:

- WebLogic Enterprise Connectivity (WLEC)
- ActiveX Client Support
- TxRPC (DCE) support

  In BEA Tuxedo 8.1, support for TxRPC (DCE) is deprecated. The Tuxedo CORBA programming interface supports an Interface Definition Language based on the DCE IDL, so moving to the Tuxedo CORBA interface is an option for customers currently using TxRPC. Another option is to code your applications using ATMI.
- CORBA Java Client

  In BEA Tuxedo 8.1, support for CORBA Java client is deprecated. Customers are encouraged to migrate to WebLogic Server 8.1 ORB or Sun Java ORB.

Online Documentation

The BEA Tuxedo product documentation is available from the following locations:

On the BEA Systems, Inc. corporate Web site. From the BEA Systems Home page at `http://www.bea.com`, click on Product Documentation, or go directly to the BEA Tuxedo "e-docs" product documentation page at `http://edocs.bea.com/tuxedo/tux81/index.htm.`

- On the BEA Tuxedo Documentation CD-ROM. The documentation CD-ROM includes Web-browsable HTML and easy-to-print Adobe Acrobat PDF documentation for this product.

Accessing the Documentation CD-ROM on Microsoft Windows Systems

To access the online documentation on CD-ROM, proceed as follows:

1. Insert the BEA Tuxedo online documentation CD-ROM into the drive.
2. Using Windows Explorer, click `index.htm` in the following directory of the Online Documentation CD-ROM:

   ```
   docs\tuxedo\tux81\index.htm
   ```

   The documentation home page is displayed in your browser.

Accessing the Documentation CD-ROM on UNIX Systems

To access the online documentation, proceed as follows:

1. Insert the BEA Tuxedo online documentation CD-ROM into the drive.
2. Mount the CD-ROM. This step might not be required for Solaris systems.

Note: For mounting instructions for the supported UNIX platforms, refer to Appendix A of *Installing the BEA Tuxedo System*.

3. Start the Netscape browser and set the browser to `/mnt/docs/tuxedo/tux81/index.htm` (where `mnt` is the CD-ROM mount point) and press Enter.

The documentation home page is displayed in your browser.

Accessing the Java API Documentation

Documentation for the BEA Jolt 8.1 is automatically installed on each machine on which the BEA Tuxedo software has been installed. This is in addition to the Java API documentation available on the online documentation CD-ROM.

The BEA Tuxedo API documentation is installed in the following location. `TUXDIR` represents the top-level directory where BEA Tuxedo is installed:

On Windows Systems:

For BEA Jolt: `%TUXDIR%\tuxedo8.1\udataobj\jolt\doc\index.html`

On UNIX Systems:

For BEA Jolt: `$TUXDIR/tuxedo8.1/udataobj/jolt/doc/index.html`

Using a Netscape or Microsoft Internet Explorer browser, open the `index.htm` start page in that directory.

Copying the Product Documentation to Your System

Although it is not necessary to copy the online documentation to your system, you can do so. Depending on the speed of your computer, you may want to copy the content of the CD-ROM to a local drive for better response time.

To make the content of the CD-ROM available on the network, put the CD-ROM in a server's CD-ROM reader and designate it as shared. This option is an alternative to taking up approximately 100 MB on the server's hard drive.

Another option is to copy the contents of the Documentation CD-ROM to a Web server on your corporate intranet. For more information about using a Web server, see the section “Microsoft Internet Explorer 5.0 Bug and Patch” on page 79.

Copying the Product Documentation to Microsoft Windows Systems

To copy the content of the online documentation CD-ROM to your system, proceed as follows:

1. Insert the online documentation CD-ROM into the CD-ROM reader.
2. Using Windows Explorer, double-click the CD-ROM drive icon. Windows Explorer displays a `doc/` folder at the root of the CD-ROM.
3. In the top directory, select the `doc/` folder and copy the content of the CD-ROM using Edit—>Copy on the Windows Explorer menu bar, or press Ctrl+c.

4. Paste the copy on your local drive (for example, your C: drive) using Edit—>Paste on the Windows Explorer menu bar, or press Ctrl+v.

Note: The content of the CD-ROM can also be copied by using the drag-and-drop feature of Windows Explorer.

Copying the Product Documentation to UNIX Systems

To copy the content of the Online Documentation CD-ROM to your system, proceed as follows:

1. Mount the CD-ROM. This step might not be required for Solaris systems.

Note: For mounting instructions for the supported UNIX platforms, see *Installing the BEA Tuxedo System*.

2. Change directory to the target directory where you want to place the files. For example: `cd /mydirectory/docs`.

3. Copy the entire CD-ROM content using a recursive copy command. For example:
   ```
   cp -r /mnt/cdrom/*.
   ```

Printing from a Web Browser

You can print a copy of the online documents, one file at a time, from your Web browser. Before you print, make sure that the chapter or appendix you want to print is displayed and *selected* in your browser. We recommend the Adobe Acrobat PDF format as a better format for printing hard copies of the BEA Tuxedo documentation, instead of printing HTML files from the browser.

Printing Adobe Acrobat PDF Files

The BEA Tuxedo documentation also includes Adobe Acrobat PDF files of all the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document, as follows:

1. On the online documentation home page, click the PDF Files button.

2. Scroll to the entry for the document you want to print.

Product Constraints

Table 5 describes product constraints for BEA Tuxedo 8.x and provides information about recommended workarounds.

Table 5 Product Constraints

1.	**Reserved Repository IDs.**	
	Description	Use of `Repository` as an interface repository ID in a `#pragma` ID OMG IDL directive, or use of `pk_` as the beginning part of such an ID, causes a conflict with IDs used internally in the BEA Tuxedo system. Their use results in undefined behavior.
	Platforms	All.
	Workaround	If you use the `#pragma` ID OMG IDL directive, do not use `Repository` or any identifier that begins with `pk_` as the repository identifier.
2.	**Restrictions to using the Dynamic Invocation Interface (DII).**	
	Description	When you use DII, you may encounter problems if you do not observe certain restrictions.
	Platforms	All.

Table 5 Product Constraints (Continued)

	Workarounds	Adhere to the following restrictions: ■ If you use `CORBA::Request::set_return_type()`, you must set the location where you want the ORB to place the result for the specified request. This can be done by using `CORBA::Any::replace()` (after setting the return type) on the Any reference returned by `CORBA::Request::return_value()`. ■ The CORBA 2.2 specification appears to imply that when using DII the user need not specify the return type or any `out` only arguments; or, if specified, the user need not specify the location in which to store the result or `out` values. In any case, the BEA Tuxedo software requires that you specify the return type, any `out` only arguments, and the location in which to store the result or `out` values. ■ When using `CORBA::ORB::send_multiple_requests_deferred()`, if any of the individual requests results in an error (that is, throws a locally detected exception), the exception is propagated to the user and any requests in the sequence after the said request will not be sent (that is, the behavior is as if the `CORBA::INV_TERM_ON_ERR` flag had been specified). The workaround is to not use this function; instead, call `CORBA::Request::send_deferred()` on each request in the sequence with appropriate `try/catch` statements around each call.
3.	**Remote BEA Tuxedo client logon/authenticate binary userdata cannot contain NULL's.**	
	Description	With an AuthType level of `Tobj::TOBJ_APPAUTH`, remote BEA Tuxedo client binary user_data passed into the logon/authenticate API cannot contain NULL's.
	Platforms	All.
	Workaround	It is possible for binary user_data to contain legitimate NULL's. One solution is to use the native mode rather than the IIOP mode. The native mode does not have the problem.
4.	**Client invocation times out after 60 seconds by default.**	
	Description	The default behavior for a client application invoking a BEA Tuxedo IIOP Server Listener/Handler or server application is to time out if 60 seconds elapse without a response. The client application receives a `NO_RESPONSE`, `NO_RESOURCES`, or `COMM_FAILURE` exception.
	Platforms	All.
	Workaround	The default timeout can be changed for the application by adjusting `SCANUNIT` and `BLOCKTIME` in the `UBBCONFIG` file for the application, or by decreasing the load on the server.
5.	**Known problem with fonts when running the University sample applications.**	

Table 5 Product Constraints (Continued)

	Description	When running the University sample applications, the availability and size of fonts varies from platform to platform and even from machine to machine, depending on the installation. As a result, text sizes may appear too large or too small on some platforms.
	Platforms	All.
	Workaround	None.
6.	**Transactions with deferred, synchronous requests experience problems.**	
	Description	When using DII, if you initiate deferred synchronous requests in the context of a transaction, the transaction does not complete successfully.
	Platforms	All.
	Workaround	Wait until you have received responses for deferred synchronous requests before you commit the transaction. Otherwise, results are unpredictable.
7.	**Microsoft's Visual C++ 6.0 compiler does not evaluate certain `long minimum` values correctly.**	
	Description	Although the OMG specification for integer literals specifies the `long minimum` value to be -2^{31}, the MSVC 6.0 compiler does not evaluate this expression correctly when it is defined in the generated code as -2147483648. This is because the positive numeric value is evaluated first and found to exceed the `long MAXIMUM` value of 2147483647, effectively reducing the `long minimum` value to -2147483647.
	Platform	Microsoft Windows NT.
	Workaround	Do not define `long minimum` literals with a minimum value of -2^{31}.
8.	**C++ reserved words in IDL for operation names.**	
	Description	Do not use C++ reserved words as operation names in IDL files. Using C++ reserved words as operation names might cause the IDL compiler to fail.
	Platforms	All.
	Workaround	Change an operation name in the IDL file to a name that is not a C++ reserved word.
9.	**Mapping of CORBA and FML data types for applications that allow interoperability between BEA Tuxedo ATMI and BEA Tuxedo CORBA.**	

Table 5 Product Constraints (Continued)

Description	BEA Tuxedo applications use FML field buffers that support a limited number of data types as described in the *Programming BEA Tuxedo ATMI Applications Using FML*. If you want BEA Tuxedo and BEA Tuxedo CORBA applications to interoperate, you must handle the mapping between the corresponding data types carefully. For example, a `CORBA::Long` is not a long for all platforms. Hence, there needs to be a conversion when reading from or writing to FML buffers after and prior to doing tpcalls from BEA Tuxedo applications. If a BEA Tuxedo application were to make a BEA Tuxedo call passing a `CORBA::long` as an FML long, a straight forward use would be as follows: Fadd32(fmlbuf, LongFmlFld, 0, (char*) &WLE_CORBA_long, 0); which you would issue prior to doing a tpcall such as: tpcall((char*)SomeTuxService, (char*) fmlbuf, (char**) &fmlbuf, sizeof(fmlbuf), 0) Such a call will succeed on most platforms, but will fail on Compaq Tru64 UNIX. **Note:** Any applications that use the BEA Tuxedo Notification Service to allow interoperability between BEA Tuxedo events and BEA Simple or CosNotification Structured Events would be affected by this problem.
Platforms	All.

Table 5 Product Constraints (Continued)

Workaround For BEA Tuxedo and BEA Tuxedo CORBA applications to interoperate, you must change the Fadd32 statement from:

```
Fadd32(fmlbuf, LongFmlFld, 0, (char*) &WLE_CORBA_long, 0);
```

to:

```
 long FML_long = WLE_CORBA_long;
  Fadd32(fmlbuf, LongFmlFld, 0, (char*) &FML_long, 0);
```

A reverse conversion would likewise be required to receive a `long` value returned from a `tpcall` and used as a `CORBA::Long`.

The University Wrapper sample application shows an example of this, where the `ACCOUNT_NO` is a `long` in the BEA Tuxedo Billing application wrapped by the BEA Tuxedo `billw_server`. The BEA Tuxedo `univw_server` and `billw_server` use a `CORBA::Long` for the `AccountNumber` and handle the conversion appropriately as follows:

```
CORBA::Double Teller_i::get_balance(
BillingW::AccountNumber account)
   {
    long account_l = account;
           :
    call_tux(m_tuxbuf, "CURRBALANCE");
           :
    }
```

10. The C++ IDL compiler does not support that valuetype “supports” regular interface or abstract interface.

Table 5 Product Constraints (Continued)

Description The generated code by C++ IDL compiler for valuetype “supports” regular interface or abstract interface is incorrect. For example:

```
// IDL
   interface ISimple {
       ...
   };

   abstract interface IAbsSimple {
       ...
   };

   valuetype A supports ISimple {
       ...
   };

   valuetype B supports IAbsSimple {
       ...
   };
```

The generated code for valuetype A and B are incorrect.

Platforms All.

Workaround Avoid defining valuetype supports regular interface or abstract interface.

11. The C++ IDL compiler does not generate correct codes for _copy_value() method if defined valuetype graph contains cycles.

Description The CORBA OMG specification is vague about how to avoid infinite loops when copying a valuetype graph that contains cycles. For example:

```
//IDL
   valuetype A;
   valuetype B {
       public A foo;
   };
   valuetype A {
       public B bar;
   };
```

If the _copy_value() function of valuetype A is invoked, it will call _copy_value() function of valuetype B, while _copy_value() function of valuetype B will call _copy_value() function of valuetype A, then infinite loops are in the _copy_value() function when invoked.

Platforms All.

Table 5 Product Constraints (Continued)

	Workaround	Override the _copy_value() function and make it so it can deal with the loops in a manner specific to the valuetype involved or avoid defining a valuetype graph that contains cycles.
12.	**The C++ IDL compiler does not support using C++ keyword as an identifier for valuetype.**	
	Description	The generated code by the C++ IDL compiler for valuetype that includes C++ keywords as an identifier is incorrect.
	Platforms	All.
	Workaround	Avoid using C++ keywords as the identifier for valuetype.
13.	**The pure virtual accessor and modifier functions for valuetype private state members are not protected.**	
	Description	Tuxedo IDL compiler does not comply with CORBA specification when compiling valuetype with private state members. CORBA specification requires that the private members' pure virtual accessor and modifier functions should be protected, but the generated pure virtual accessor and modifier functions by Tuxedo IDL compiler are public. For examples:

```
//IDL
module A {
    valuetype V{
        ......
        private V m_v;
        ......
    };
};


// The generated code:
namespace A{
    class V : public virtual ::CORBA::ValueBase
    {
        ...
        public:
            // accessor and modifier for m_v
            virtual ::A::V * m_v() const = 0;
            virtual void m_v(::A::V *) = 0;
    }; // class V
}
```

	Platforms	All.

Table 5 Product Constraints (Continued)

Workaround None.

14. CORBA Object-By-Value semantics are not supported in "InProc" invokes.

Description

If the CORBA server acting as a client invokes an IOR for an object that is activated in the same server process, then this is considered an "InProc" invokes and the method is invoked directly with no GIOP message generated.

The Object-By-Value semantics are not supported in "InProc" invokes scenario. If valuetype/valuebox is passed as an argument or returned as a return value in "InProc" invokes scenario, the CORBA::INTERNAL exception will be raised.

For example:

```
valuetype A {
    ...
};
interface foo {
    void op1(in A arg1);
};
interface bar {
    void op2(in foo arg2);
};
```

If the op2() method of "bar" object invokes op1() method of "foo" object passed in, and both "foo" and "bar" objects are deployed in the same server process, a CORBA::INTERNAL exception will be thrown.

Platforms All.

Workaround Avoid passing valuetype/valuebox in "InProc" invokes scenario, or deploy CORBA objects in different server processes.

15. Valuetypes that formed circular graphs will cause a CORBA server memory leak.

Table 5 Product Constraints (Continued)

Description	CORBA specification allows that Valuetypes can be used to form arbitrary, potentially circular graphs. But the reference counts of circular Valuetypes may never drop to zero, this will cause server memory leak. CORBA specification is vague about how to resolve this problem in C++. BEA Tuxedo CORBA does not provide a mechanism to resolve such problems. This means if Valuetypes formed circular graphs, the CORBA server will have memory leak.
	For examples:
	`// IDL`
	`valuetype A;`
	`valuetype B;`
	`valuetype A {`
	`    public B foo;`
	`}`
	`valuetype B {`
	`    public A bar;`
	`}`
	Valuetype A and B formed a circular graph.

Platforms	All.
Workaround	Avoid defining Valuetypes that will form circular graphs.

Known Problems in BEA Tuxedo 8.1

The following sections describe known problems with the BEA Tuxedo software and include recommended workarounds. The problems are listed by the Change Request (CR) number. The CR number is provided to facilitate the tracking of these problems.

Contact your BEA Customer Support Center for assistance in the tracking of any unresolved problems. When contacting the BEA Customer Support Center, please refer to the CR number.

Table 6 lists known problems for BEA Tuxedo 8.1.

Table 6 Known Problems for Tuxedo 8.1

CR012652	**Sequences and arrays contained by a `CORBA::Any` cannot be passed remotely due to their lack of a repository ID.**	
	Problem	Sequences and arrays contained within a `CORBA::Any` cannot be passed remotely. These types do not have a repository ID in the syntax specified for transferring them across the wire. For a description of the Common Data Representation (CDR) Transfer Syntax, see Table 13-2 of *The Common Object Request Broker: Architecture and Specification*, Revision 2.2. The repository ID is used by the receiving ORB to unmarshal the type contained by the `Any`. Without the ID, the ORB cannot obtain sufficient information to unmarshal the data type. This restriction may be removed in a future release.
	Platforms	All.
	Application Type	CORBA C++.
	Workaround	Sequences and arrays can be used within other data types, such as structures, or they can be used directly as parameters.
CR012697	**When you use VIEWs, strings are null terminated and characters truncated.**	
	Problem	There is an inconsistency between BEA Jolt and native BEA Tuxedo clients. BEA Jolt does not know what a client and server agreed to exchange—a string or a null-terminated string.
	Platforms	All
	Workaround	Accommodate a null in the definition of the string.
CR013809	**Administration Console: Select FIle/Exit from Netscape caused a termination.**	
	Problem	Using Netscape 4.0.5: 1. Start Administration Console. 2. Bring up online help. 3. Select File/Exit from Netscape which shows online help. It terminates Netscape where Administration Console starts.
	Platforms	Sun Solaris

Table 6 Known Problems for Tuxedo 8.1

	Workaround	This is a vendor problem. Netscape does not pop up an alert dialog when exiting on Solaris even if there are multiple Netscape windows open.
CR014128	**ISL will not start if there is an underscore (_) in the IP name.**	
	Problem	The ISL fails to boot if the `-n` option has a hostname that contains an underscore. The error written in the user log (ULOG) is: `ISNAT_CAT:1242: ERROR: Bad Internet type of listening address provided: <node name>`
	Platform	Microsoft Windows.
	Workaround	Ensure that the ISL is configured correctly with host and port values.
CR016275	**Cannot have two data types for the same field name in input and output VIEWs.**	
	Problem	The member in a VIEW is qualified by the structure name, so the input view and output view may use the same name for a member, but with different data types. However, BEA Jolt does not support this feature and some existing BEA Tuxedo services cannot be accessed by BEA Jolt client. A typical service that uses this feature is a gateway; it translates one data type to another data type with the same name.
	Platforms	All

	Workaround	Use the `VIEW` name to qualify the field name. For example, INVIEW and OUTVIEW both have a field birthdate, but one is an integer and the other is a string.
		`svc.setlnt[“INVIEW.birthdate”,19980308];`
		`svc.call[null];`
		`String bdate = svc.getStringDef[“OUTVIEW.birthdate”,null]`

CR017633	**User’s `jrepository` file needs to be converted to use the services in JREPSVR.**	
	Problem	Because the JREPSVR in BEA Jolt 1.1 has been changed to use FML32 instead of FML16, BEA Jolt 1.1 users should update their BEA Jolt Repository file (`jrepository`) to use the services in JREPSVR. This buffer type update is recommended, but not required because the old FML16 buffer type can still work with the new JREPSVR.
	Platforms	All

Table 6 Known Problems for Tuxedo 8.1

	Workaround	BEA Jolt 1.1 user should update the BEA Jolt Repository file (`jrepository`) to use the services in JREPSVR.
CR017675	**Changing FML data types does not update JSH with changes.**	
	Problem	It does not appear that after changing the FML field data type from `int` to `string` the JSH gets updates.
	Platforms	All
	Workaround	After you change the FML field definitions, reboot JSL/JSH.
CR018177	**client_response() or target_response() method.**	
	Problem	Invocations on a `CORBA::Object` from a C++ interceptor in the `client_response() or target_response()` method is not supported. The results of such an operation are not predictable.
	Platforms	All.
	Workaround	None.
CR018211	**RLI administration restrictions.**	
	Problem	The RLI administrative functions of registering, unregistering, and changing interceptor order are not supported while the BEA Tuxedo application is running. The RLI administrative functions must only be done before issuing the `tmboot` command or after issuing the `tmshutdown` command. BEA Tuxedo application results are not predictable if interceptors are administered while the application is running.
	Platforms	All.
	Workaround	Do not register, unregister, or change the interceptor order while the BEA Tuxedo application is running.
CR018476	**Missing host:port on ISL -n option causes Application and TMNTS servers to crash.**	
	Problem	If you define an ISL without a `-n host:port` entry in a `UBBCONFIG` file, an application server that uses the notification server (TMNTS) and TMNTS crashes.
	Platforms	All.

Table 6 Known Problems for Tuxedo 8.1

	Workaround	According to RFC 810, the document that specifies the syntax for IP hostnames, hostnames cannot contain an underscore (_) character. Therefore, a listening address such as `//my_computer.com:8000` is considered invalid and processes such as ISL and `tlisten` fail to start. This is a problem on Microsoft Windows NT systems because hostnames with underscore characters are often used. If you have a Windows NT system whose hostname contains an underscore, do one of the following: ■ Rename the machine so that its hostname does not contain an underscore, and change the corresponding name in the Domain Naming System (DNS) files or, if you are not using DNS, change the corresponding name in all the local HOST files. ■ Add a DNS alias (without an underscore) to the machine by adding a `CNAME` record. ■ Add an alias to the `%WINDIR%\System32\drivers\etc\hosts` file: `123.45.67.89 my_computer.com mycomputer.com` where the first entry is the IP address of the machine. Then perform the following procedure: 1. Reboot the machine. 2. In the `UBBCONFIG` file, use `my_computer.com` when specifying the `PMID` entry for this machine in the `MACHINES` section. 3. In all other places in the `UBBCONFIG` file, such as the `NETWORK` or `SERVERS` sections, or when starting listening processes, such as `tlisten` on the command line, use `mycomputer.com` as the hostname.
CR018811	**Dynamic updating of the Domains configuration file (`dmadmin -c`) is not supported for changes to the BEA TOP END password field in the `DM_TOPEND` section. The Domains MIB does not support dynamic reconfiguration of the password, either.**	
	Problem	Dynamic updating of the Domains configuration file (`dmadmin -c`) is not supported for changes to the BEA TOP END password field in the `DM_TOPEND` section. The Domains MIB does not support dynamic reconfiguration of the password, either. The `GWADM` server ignores the password field on incoming requests to add or update entries in the `DM_TOPEND` section of the domains configuration. Any dynamic update requests from either the `dmadmin` command or the Domains MIB will not have the desired effect with regard to changing the BEA TOP END password.

Table 6 Known Problems for Tuxedo 8.1

	Platforms	All.
	Workaround	To update the password for a `DM_TOPEND` entry, run the `dmadmin topendpasswd` command. For more information, see `dmadmin`(1).
CR019447	**Signatures get dropped when floating point values are used.**	
	Problem	When a process is determining whether to keep or disregard a previously attached signature on a message buffer, it recalculates the signature on that buffer. If the new signature does not match the old one, then the previously verified signature on the buffer is silently dropped by the system. The behavior of doubles and floats in this context may cause problems. Signatures are calculated on the encoded representation. For buffer types supplied by the BEA Tuxedo system, this calculation causes doubles and floats to be encoded as XDR doubles. XDR doubles may not have the same precision as the native doubles or floats on your machine. Therefore, when a buffer containing a floating-point number or double precision floating-point number is transported to a machine with different floating-point precision and decoded on that machine, the resulting value may differ from the original value. When the decoded number is then re-encoded from native format to XDR format, the encoding may be different. Therefore, the signature will not be verified and will be silently dropped, even if the application did not change the buffer at all.
	Platforms	All.
	Workaround	If you are using buffer types that are not supplied by the BEA Tuxedo system, replace the `_tmencdec` function in the types switch to use an encoded format with as much precision as the native format (everywhere).
CR019611	**`tpcall()` fails when it is invoked (with the `TPNOCHANGE` flag set and FML or FML32 buffers) by clients on pre-Release 7.1 nodes.**	
	Problem	In an MP configuration running multiple releases of the BEA Tuxedo system, a native client running on a release 6.5 (or earlier) node invokes `tpcall()` for a service on a release 7.1 node. If `tpcall()` sends an FML or FML32 buffer and the `TPNOCHANGE` flag is set, the call fails with `tperrno` set to `TPEOTYPE`. This problem does not occur for Workstation clients on pre-release 7.1 nodes.
	Platforms	All.

Table 6 Known Problems for Tuxedo 8.1

	Workaround	Do not set the `TPNOCHANGE` flag in `tpcall()` when pre-release 7.1 native clients are calling, with FML buffers, services on a release 7.1 node.
CR019629	**If `BUFTYPE` is not set to `ALL`, calls from pre-Release 7.1 clients to Release 7.1 services with `tpcall()` fail.**	

	Problem	If, in a mixed-release environment, a pre-release 7.1 client calls a release 7.1 service and the `BUFTYPE` parameter is not set to `ALL` (a default if not specified) for this service, then `tpcall()` fails with `tperrno` set to `TPEITYPE`.
		This problem occurs even when the caller's buffer type matches the specification for the service in the `UBBCONFIG` file. For example, suppose a release 7.1 service called `TOUPPER` is specified in `UBBCONFIG` as follows:
		`*SERVICES`
		`TOUPPER BUFTYPE=STRING`
		A call to a `TOUPPER` service with a `STRING` buffer from a release 6.5 client will still fail.

	Platforms	All.
	Workaround	If your application offers services from servers on a release 7.1 node, when you create entries for them in the SERVICES section of the UBBCONFIG file, take one of the following actions: ■ Do not set the BUFTYPE parameter ■ Set the BUFTYPE parameter to ALL
CR019848	**The TOP END Domain Gateway (TEDG) does not correctly support loop-back transactions through the same gateway instance in failure situations.**	
	Problem	When TOP END Domain Gateway is used in the following scenario: 1. A client in one domain transactionally invokes a service in another domain. 2. The invoked service, in turn, invokes a service in the originating domain. If a failure occurs during this type of scenario, the transaction may not be completely rolled back. It is possible that certain resource manager participants may not receive a rollback notification, leaving the transaction in an inconsistent state.
	Platforms	All platforms for which the TOP END Domain Gateway is supported

Table 6 Known Problems for Tuxedo 8.1

	Workaround	Avoid loopback configurations of this type; currently, they are not supported by the TOP END Domain Gateway.
CR019994	**Release 7.1 Workstation clients fail to connect to a Release 6.3 node.**	
	Problem	When a release 7.1 Workstation client tries to connect to a release 6.3 node, and the Workstation link is configured for 56-bit LLE, `tpinit()` fails with `tperrno` set to `TPESYSTEM`. The error message in the user log is: `LIBWSC_CAT:1357: ERROR: Received message not intended for this client`
	Platforms	All.

	Workaround	If 56-bit LLE or 128-bit LLE is enabled in a release 7.1 environment, then one of the following sets of environment variables must be set:
		1. `TMMINENCRYPTBITS=0`
		`TMMAXENCRYPTBITS=0`
		2. `TMMINENCRYPTBITS=0`
		`TMMAXENCRYPTBITS=40`
		3. `TMMINENCRYPTBITS=40`
		`TMMAXENCRYPTBITS=40`
		4. `TMMINENCRYPTBITS=128`
		`TMMAXENCRYPTBITS=128`
		If `TMMINENCRYPTBITS` and `TMMAXENCRYPTBITS` are set to any other values for a 56-bit LLE environment, Workstation client connections to release 6.3 nodes will fail.

CR020115	**A BEA TOP END client cannot enqueue a zero-length buffer to a BEA Tuxedo /Q queue via the TOP END Domain Gateway.**	
	Problem	The BEA Tuxedo /Q component does not properly interpret buffers of zero length. Therefore, when a BEA TOP END client attempts to enqueue a zero-length buffer to a BEA Tuxedo /Q queue, the BEA Tuxedo queue manager drops the message because it is of an unknown type.
	Platforms	All platforms for which the TOP END Domain Gateway is supported
	Workaround	Always enqueue a non-zero length buffer.
CR020175	**Automatic calls to `tpterm()` may not work for multithreaded and multicontexted clients.**	

Table 6 Known Problems for Tuxedo 8.1

	Problem	The BEA Tuxedo system automatically calls `tpterm()` if a client exits without calling it explicitly. In a multicontexted and multithreaded environment, however, the release 7.1 system cannot detect the number of contexts that are active when the client terminates; only a single call to `tpterm()` is made. As a result, only one context is shut down. Other limitations, especially when unsolicited thread notification is used, may also exist.
	Platforms	All.
	Workaround	Make sure that before a multicontexted client exits, it explicitly calls `tpterm()` for all active contexts.
CR020428	**When encryption is used, the BEA Administration Console client applet does not work with Netscape 4.7 on HP-UX 11 with HP Java Plug-in (JPI) installed.**	
	Problem	The client-side applet of the BEA Administration Console cannot be used with Netscape 4.7 on an HP-UX 11 with the HP Java Plug-in (JPI) installed.
	Platforms	HP-UX 11.0.
	Workaround	Currently, there is no workaround for using the BEA Administration Console client-side applet on an HP-UX 11 machine. To use the client-side applet, run it on a Solaris or NT desktop with Netscape 4.6.1, Netscape 4.7, or Internet Explorer 5.0. Netscape is working on a fix to its 4.7 browser and HP is working on a fix for Java Plug-in for HP-UX 11. Once these fixes become available, you will be able to run the client-side applet of the BEA Administration Console on HP-UX 11 with a Netscape browser.
CR020514	**There is a 4K limit on authentication tokens on `GWTDOMAIN` links.**	
	Problem	The amount of data that can be sent in authorization tokens over `GWTDOMAIN` links may not exceed 4000 bytes. In other words, the amount of data cannot exceed 4000 bytes when the data is being returned to the caller through one of the following plug-in call-back methods: ■ `_ec_sec_gss_init_sec_context` ■ `_ec_sec_gss_accept_sec_context` These methods are registered under the selector `gwtdomain/security/authentication`.
	Platforms	All.

Table 6 Known Problems for Tuxedo 8.1

	Workaround	There is no workaround for a security provider that needs tokens larger than 4000. A security provider may be able to compress tokens.
CR020908	**The BBL process in BEA Tuxedo dumps core at master site of a MP mode application.**	
	Problem	This problem is caused by system's particular MSGQ IPC resource configuration. The MSGMNB is 14336000 and MSGMAX is 131072. In `tmmbsendm()` we call `msgctl()` to get the state of the BBL's message queue, and 14336000 is returned as part of the query. So, it should be safe to send the reply through message queue. Normally `msgsnd()` will fail when the message is too large and doing file transfer, but in this case it succeeded, `msgsnd()` returns 0. What BBL received is a corrupted message, and caused the `SEGV` exception in `_tmfmsg_postrecv()`.
	Platforms	Solaris.
	Workaround	When MSGMNB and MSGMAX is modified to 65536, it works. Modifying MSGMNB to 131072 also works.
CR025411	**SET operation fails to reset the state of the class from ACTIVE to OPEN.**	

	Problem	The `SET` operation is as follows:
		`TA_OPERATION    SET`
		`TA_CLASS                T_APPQSPACE`
		`TA_APPQSPACENAME        MYQSPACE1`
		`TA_QMCONFIG     D:\temp\tmp.1\files/QUE1`
		`TA_LMID L1`
		`TA_STATE        OPE`
		The output does not show the `TA_STATE` as `OPEN`, instead it shows it as `ACTIVE` only.
		Note: This happens for `tpadmcall` only. The `SET` operation passes for `tpcall`, `tpacall` and `tpenque`.

	Platforms	Microsoft Windows 2000.
	Workaround	Use `tpcall` or `tpacall` instead of `tpadmcall`.
CR031385	**Value of `TA_CURDISPATCHTHREADS` field in `TA_SERVER` class in not updated properly.**	

Table 6 Known Problems for Tuxedo 8.1

	Problem	The value of the TA_CURDISPATCHTHREADS field in the TA_SERVER class is not updated properly on Windows 2000 whenever there is a new thread started for the server.
	Platforms	Microsoft Windows 2000.
	Workaround	None.
CR031816	**Requests on methods using multithreading appear to time out prematurely.**	
	Problem	In some timing situations involving multithreaded applications, a method can time out too early. A symptom of this problem is an error message similar to the following in the log file: `LIBTUX_CAT:669 "ERROR: Message operation failed because of the invalid message queue identifier."` Note that this error message can also result from insufficient configuration settings for options such as MAXACCESSORS and MAXWSCLIENTS. Verify these settings before increasing the timeout settings.
	Platforms	All.
	Workaround	Increase the UBBCONFIG file settings for BLOCKTIME and SCANUNIT. For example: `SCANUNIT 5, BLOCKTIME 6000`.
CR036436	**BEA Tuxedo 8.1 requirement for interoperability with WLE 5.0.1**	
	Problem	WLE 5.0.1 does not support AAA_TCM. Any op codes coming from DBBL will have AAA_TCM. This causes the BBL to fail in WLE 5.0.1.
	Platforms	All.
	Workaround	For interoperation between BEA Tuxedo 8.1 and WLE 5.0.1, export the environment variable ALLOW_INTOP on the MASTER side.
CR039550	**Memory leak using poll_next_response() for native client.**	
	Problem	When running a CORBA client in native mode, calls to `poll_next_response()` create a small memory leak. Because `poll_next_response()` is typically invoked iteratively, this may eventually result in a NO_MEMORY exception.
	Platforms	All.

Table 6 Known Problems for Tuxedo 8.1

	Workaround	One solution is to use IIOP mode rather than native mode. IIOP mode does not have the problem. Another is to insert a short pause in the iterative loop so that the leak does not expand too quickly. Even a 10 millisecond pause in the loop will greatly slow down the leaking. As large a pause as is acceptable is best.
CR041877	**`tmboot` reports "`executable file not found`" error.**	
	Problem	BEA Tuxedo 8.1 on Windows 2000 (with service pack 1 installed) has a problem during `tmboot` accessing the executable when it is on a shared network drive. The drive type is this scenario is a Samba mounted drive. Although the file is accessible on the Windows 2000 test machine, from the DOS prompt or through the Windows Explorer, `tmboot` reports that it is unable to execute the file because the file was not found.
	Platforms	Microsoft Windows 2000 with service pack 1.
	Workaround	Uninstall service pack 1 on Windows 2000.
CR041956	**Administration console can be launched using IE 4.0/5.0 but with some performance issues.**	
	Problem	The Administration Console can be launched using Internet Explorer 4.0 and 5.0, but it just takes a long time to load the applet. On the average, it takes approximately 15 to 45 minutes to load the applet.
	Platforms	All.
	Workaround	None.
CR042034	**ud32 crashes when `SYSTEM_ACCESS=PROTECTED` and with `-C tpsysadm.`**	
	Problem	The ud32 `-C tpsysadm` crashes when `SYSTEM_ACCESS` in the configuration file is set to `PROTECTED`.
	Platforms	All.
	Workaround	None.
CR042294	**ACL Security Failure on `tpconnect` does not return disconnect to OSI-TP Gateway.**	

Table 6 Known Problems for Tuxedo 8.1

	Problem	Whenever a `tpconnect (TPSENDONLY)` is issued by a client to a service for which the ACL Security fails (client user ID not found in a group for which the service is allowed), a `nw_disconnect` is not returned to the OSI TP 4.0 Gateway. The following error messages are output to the ULOG:
		`111324.DALNTSMP44!CONVSERV.240.310.0:`
		`LIBTUX_CAT:1515: WARN: Access control violation -`
		`user 2 on SITE1 tried to access SERVICE TOUPPER`
		`111324.DALNTSMP44!CONVSERV.240.310.0:`
		`LIBTUX_CAT:6187: WARN:`
		`AUDIT_POSTOP`
		`SECURITY FAILURE: who = , operation_name = CONNECT`
		`OPERATION,`
		`operation_target = TOUPPER`

	Platforms	All.
	Workaround	If it is a valid client, create the ACL entry.
CR044429	**Dmloadcf dumps core when loading over an existing bdmconfig file.**	
	Problem	If you do not remove the `bdmconfig` file, `dmloadcf` dumps core, even when reloading the exact same `dmconfig` file.
	Platforms	All.
	Workaround	Remove the `bdmconfig` file before doing `dmloadcf`.
CR045148	**Tlisten has a typo on its usage.**	

	Problem	The usage of tlisten shows as:
		`tlisten -l naddr [-d dev] [ -u {uid-# |uid-name}]`
		`[-z {0|40|56|128}]`
		`[-Z {0|40|56|128}]`
		`-l naddr` should be `-l nlsaddr` as it appears in the document. This typo misleads the user to enter in an incorrect input.

	Platforms	All.

	Workaround	Use this usage instead:
		`tlisten -l nlsaddr [-d dev] [ -u {uid-# | uid-name}]`
		`[-z {0|40|56|128}]`
		`[-Z {0|40|56|128}]`

Table 6 Known Problems for Tuxedo 8.1

CR045947	**NO_MEMORY exception on HP.**	
	Problem	Testing of the Active Object Map (AOM) limit for 100,000 active objects gets the `NO_MEMORY` exception that reports insufficient physical memory. The HP system, on which the test fail, had about 850 MB of physical memory. The Solaris system, which had 1.025 GB, worked okay for the test. The same test on Tru64 gets `INTERNAL` exception while creating the 33 thousandth active object.
	Platforms	HP-UX and Tru64.
	Workaround	None
CR046254	**TA_DMCODEPAGE returns garbage.**	
	Problem	A DMIB test case tries to create a remote service entry using ud32; its TA_DMCODEPAGE was *not* passed in as part of the input. The operation succeeds without any issues, but in the return packet, the TA_DMCODEPAGE parameter is returned as garbage.
	Platforms	All.
	Workaround	Ignore the output for TA_DMCODEPAGE in the return packet.
CR046990 (Duplicate CR087378)	**Problem with InstallAnywhere script on Tru64 Unix 5.1.**	
	Problem	No Java virtual machine could be found from your environment variable PATH.
	Platforms	DEC Alpha
	Workaround	You must install a virtual machine prior to running the installation program. Specify the following: `export PATH=/opt/java130/bin:$PATH` where `/opt/java130/bin` contains Java executables for JDK version 1.3.0.
CR049347	**`TMQFORWARD` in infinite loop when reply queue name is unprintable**	
	Problem	`TMQFORWARD` goes into an infinite loop when the reply queue name is an unprintable ASCII string. The `TMQFORWARD` process invokes the service infinitely.

Table 6 Known Problems for Tuxedo 8.1

	Platforms	Red Hat Linux 6.2.
	Workaround	Ensure that the reply queue name is composed of printable ASCII characters only.
CR050166	**C++ definitions in wrong order in IDL->C++.**	
	Problem	The Tuxedo IDL compiler sometimes writes out a client include file that has references to a base class before the base class is completely defined.
	Platforms	All.
	Workaround	Place the interfaces or value types that inherit from other interfaces or value types in separate IDL files.
CR050593, CR051024, and CR050664	**Any insertion for WStringValue not defined. Any insertion for ValueBase not defined. Cannot unmarshall sequence of Any's.**	
	Problem	The use of Any's with value types is not supported in BEA Tuxedo 8.1. Support for inserting value types into Any's, and extracting value types from Any's, will be provided in a future release.
	Platforms	All.
	Workaround	Do not use value types with Any's.
CR050406	**Generated value `type _copy_value()` routine not complete.**	
	Problem	The Tuxedo IDL compiler has the ability to create example implementations for valuetypes through the use of the `-i` command qualifier. One of the routines created for a valuetype implementation is the `_copy_value` routine. The code generated for this routine does not conform to the OMG CORBA specification in that it only does a shallow copy of the value type members. This is insufficient for complex value types.
	Platforms	All.
	Workaround	Generate the example implementation source files and then edit the `_copy_value` routine to do a deep copy of each of the value type's data members.
CR050516	**Typecodes have incompatible type for value factories.**	

Table 6 Known Problems for Tuxedo 8.1

	Problem	The definition for typecodes, for example, `_tc_MyType->id()` returns a `const char*` whereas `register_value_factory` is expecting a `char* const`. and Microsoft Visual C hangs with this message:. `TuxClient.cpp(83) : error C2664: 'register_value_factory' : cannot convert parameter 1 from 'const char *' to 'char *const ' Conversion loses qualifiers`
	Platforms	Microsoft Windows.
	Workaround	Use a temporary variable as a go-between.
CR050552	**SSL URL syntax parsing not working correctly.**	
	Problem	A URL address list containing randomized address groups followed by comma-separated lists of addresses is not being parsed correctly.
	Platforms	All.
	Workaround	Do not mix URL types in a list.
CR050760	**Tuxedo does not correctly recognize Null value types from WebLogic Server.**	
	Problem	When a BEA WebLogic Server application encodes a null value type to be sent to a Tuxedo application, it does so by mapping these null value types as CORBA Anys containing an abstract interface. BEA Tuxedo does not correctly interpret these constructs as null value types.
	Platforms	All.
	Workaround	Edit the application to eliminate the need to pass null value type instances.
CR050841	**Nested valuetypes cause IDL compiler core dumps.**	
	Problem	In IDL definitions, value types may contain references to themselves. Under some circumstances, these constructs cause the IDL compiler to fail.
	Platforms	Compaq Tru64.
	Workaround	Edit the IDL file to eliminate nested valuetype references, or execute the IDL compiler on another platform and copy the resulting C++ stubs, skeletons, and implementation source files to the Tru64 platform.

Table 6 Known Problems for Tuxedo 8.1

CR051108	**`Tlisten.pw` file not the same between BEA WebLogic Enterprise 5.1 and BEA Tuxedo 8.1.**	
	Problem	The creation of the tlisten.pw is different between BEA WebLogic Enterprise 5.1 and BEA Tuxedo 8.1. The impact to the customer is in a mixed environment when there are BEA Tuxedo 8.1 masters and BEA WebLogic Enterprise 5.1 non-masters (potentially the same thing could be true for previous versions of BEA Tuxedo ATMI). When booting, the user will get a security violation in the `ULOG` when trying to connect the other system that has the previous version. **Note:** When viewing the files, the contents may look the same, but they are not, the file termination string is different by one character. The only way to ensure they are the same is to do a copy or file transfer.
	Platforms	All.
	Workaround	To resolve this problem or prevent it from happening, a single version of the `${TUXDIR}/udataobj/tlisten.pw` file should be copied to all systems in the domain(s). This file can either be taken from the BEA Tuxedo 8.1 home directory or the previous version's home directory.
CR071576	**tlisten.pw file not the same between WLE 5.1 and Tuxedo 8.1**	
	Problem	In a mixed environment with Tuxedo 8.1 `MASTERS` and WLE 5.1 non-`MASTERS`, you will get a security violation in the `ULOG` when trying to connect the WLE 5.1 system. This same issue may be true for previous versions of Tuxedo-ATMI.
	Platforms	All.
	Workaround	Copy a single version of the `$TUXDIR/udataobj/tlisten.pw` file to all systems in the domain. You can use this file from the Tuxedo 8.1 installation or from the previous version’s home directory. When viewing the files, the content may look the same, but they are not. The file termination string is different by 1 character. The only way to ensure the files are the same is to do a copy or file transfer.
CR071624	**FML function Fgetalloc and CFgetalloc gives GPF in Windows 2000.**	
	Problem	`Fgetalloc` gives GPF when proper FBFR*, `fieldid` is passed.
	Platforms	Windows 2000 Advanced Server

Table 6 Known Problems for Tuxedo 8.1

	Workaround	The problem can be fixed in one of the following ways in `fml_nt.mak`: ■ Use buildclient directly to compile from `.c` files into `.exe` files, setting `CFLAGS` (not `CCFLAGS`) to specify any extra compiler options. ■ Add the `/MD` flag to the `cl -c` line.
CR071960	**buildclient and buildserver fails with unresolved references.**	
	Problem	`buildclient` and `buildserver` fails with unresolved references.
	Platforms	All
	Workaround	When building a client or server with the XML library (`libtxml`), export CC= the C++ compiler before using the `buildclient` or `buildserver` command.
CR088327	**Buffer encode does not pass numeric quantities larger than 32 bits.**	
	Problem	Large long values (> 32 bits) in FML or VIEW buffers cannot be sent between different 64-bit platforms. The encode/decode functions do not encode 64-bit long values. This also applies to buffers encoded as part of `tpexport()`.
	Platforms	All 64-bit platforms.
	Workaround	If the machines are the same type, make sure that `TYPE` or `WSTYPE` are set to the same string. Otherwise, avoid sending large long values between machine types.
CR088360	**Workstation client core dumps when tpcall is made without explicit tpinit.**	
	Problem	When a workstation client makes a `tpcall` or any other call to Tuxedo without calling `tpinit` explicitly, it exits abnormally with `SIGSEGV.`
	Platforms	All.
	Workaround	Do a tpinit() explicitly.
CR089272	**Jolt 8.1 does not work with WLS 6.1 SP4 when security context is enabled.**	
	Problem	An exception is sent when a client request to WebLogic Server 6.1 (when jolt pool is configured) with Jolt 8.1 and the security context is enabled.
	Platforms	All.

Table 6 Known Problems for Tuxedo 8.1

	Workaround	There were changes made to the security feature in WebLogic Server 7.0. Due to that, Jolt had to be changed to work with the new security in WebLogic Server. The result is that if security context propagation is enabled, then Jolt 8.1 clients cannot be used with WebLogic Server releases earlier than 7.0.
CR092389 (Duplicate CR080631)	**WLE 5.1 - several ISL/ISHs shutdown and restarted may lead to client not reconnecting to any ISL.**	
	Problem	Java ORB fails to reconnect to previously down ISL (conn left in cache even if failed connect).
	Platforms	Solaris 8
	Workaround	Remove IIOPConnection from ConnectionTable's cache if socket fails to connect.
CR092413	**Inconsistent number of retries to connect to remote domain.**	
	Problem	The number of retries to connect to remote domains are inconsistent when RETRY_INTERVAL value is different for different remote domains in a local domain.
	Platforms	All.
	Workaround	The retry message is only printed once in the ULOG even when MAXRETRY > 1 and multiple retries are attempted. Check the ULOG for the following: The ULOG will show “INFO: Stopped retrying domain” if the remote domain could not be reached. Otherwise, the ULOG will show “Connection established.”
CR092441	**On a 64-bit platform running Tuxedo 6.5, tppost of a VIEW or VIEW32 to Tuxedo 8.1 sometimes fails with LIBTUX_CAT:1555.**	
	Problem	On a 64-bit platform running Tuxedo 6.5, `tppost()` of a VIEW or VIEW32 buffer to a Tuxedo 8.1 node sometimes fails with: `LIBTUX_CAT:1555: ERROR: Unsolicited message encoding/decoding failed (_tmencdec(TMDECODE) tperrno=12).`
	Platforms	All.

Table 6 Known Problems for Tuxedo 8.1

	Workaround	Depending on the specific set of fields contained in the VIEW or VIEW32 buffer, the Tuxedo 6.5 node may not be allocating sufficient space to decode the buffer. The problem either always happens or never happens for a specific VIEW or VIEW32. If this problem is occurring in your application, please contact BEA Customer Support to obtain a rolling patch for the Tuxedo 6.5 node.
CR092647	**tmboot fails in MP mode configuration between Tuxedo 8.1 and WLE 5.1.**	
	Problem	`tmboot` with Master as Tuxedo 8.1 and WLE 5.1 as non-Master fails with error message in ULOG.
	Platforms	All
	Workaround	Apply the WLE 5.1 RP60 or greater to resolve this issue.
CR092679	**Installation fails on HP-UX when LANG=ja_JP.eucJP.**	
	Problem	On HP-UX the installation program does not work when you set LANG=ja_JP.eucJP.
	Platforms	HP-UX
	Workaround	Set LANG=ja_JP.SJIS and then try to run the installation program.
CR092682	**Installer does not work on Japanese Solaris box.**	
	Problem	In console mode, the installer does not work on a Japanese Solaris box. Prior to running the installation program, you must add `/usr/ucb` to the path. If `/usr/ucb` is not the first entry in the path, the following exception occurs: Invocation of this Java Application has caused an InvocationTargetException. This application will now exit.
	Platforms	All.
	Workaround	You must add `/usr/ucb` as the first entry in the path prior to running the installation program.
CR092705	**tpcall from Tuxedo 8.1 to Tuxedo 7.1 fails with TPESYSTEM error in Domain configuration.**	

Table 6 Known Problems for Tuxedo 8.1

	Problem	If connection policy is `INCOMING_ONLY` in Tuxedo 8.1 and `ON_DEMAND` on Tuxedo 7.1, all the services are shown in suspend mode, and client call fails `TPENOENT` error.
	Platforms	All.
	Workaround	This is a problem with base Tuxedo 7.1 release. You must apply Tuxedo 7.1 RP 135 or later to resolve this issue.
CR092867	**All enqueue fails after Unprivileged user enqueues between WLE 5.1 and Tuxedo 8.1, when security is set to MANDATORY_ACL.**	
	Problem	All enqueue fails after unprivileged user enqueues when security is `MANDATORY_ACL` and the configuration is MP mode. SHM mode works fine in Tuxedo 8.1.
	Platforms	All.
	Workaround	Ensure that only authorized users access services during the limited period when the application is partially upgraded from WLE 5.1 to Tuxedo 8.1.
CR092948	**For buildtuxedo to work on Windows 2000 Advanced Server, you must log out and then log in to the system.**	
	Problem	You receive an invalid SDK license error if you do not log out and then log in after `buildtuxedo` is added to the tools in MSVC GUI.
	Platforms	Windows 2000 Advanced Server
	Workaround	Log out of the system and then log in.
CR093053	**AUTHSVR fails to boot without tpusr file.**	
	Problem	On Windows 2000, AUTHSVR will not boot without a `tpusr` file.
	Platforms	Windows 2000 Advanced Server.
	Workaround	Create a `tpusr` file using `tpusradd`. If this is not desirable, then create an empty `tpusr` file.
CR093943	**tpcancel with invalid descriptor does not return TPEBADDESC in tperrno**	
	Problem	`tpcancel` with an invalid Call Descriptor is suppose to fail with `TPEBADDESC` in `tperrno`, but it returns `tperrno` as 0.

Table 6 Known Problems for Tuxedo 8.1

	Platforms	All.
	Workaround	None. Regardless of whether `tpcancel()` returns success or fails with `TPEBADDESC`, the handle passed to `tpcancel()` is invalid following the completion of `tpcancel()`.
CR094358	**During silent install, there is an error in the files copied for CORBA if install option is invalid and silent install defaults to Full Install.**	
	Problem	If an invalid option is entered in the `CHOSEN_INSTALL_SET` parameter for silent installation, the program defaults to the Full Install. However, there is an error in the CORBA files that are copied during the default installation process.
	Platforms	All
	Workaround	When using the silent install, be sure to code a valid option for the `CHOSEN_INSTALL_SET`. To determine the correct options, refer to *Installing the BEA Tuxedo System*.
CR094840	**Installation error due to potential files being corrupted.**	
	Problem	You may not have enough space in `/tmp` if you receive an error message at install that states “the size of the extracted files to be installed are corrupted.”
	Platforms	All.
	Workaround	Try to clean out `/tmp` and free some space. Attempt the installation again to determine if the error is due to space or corrupt files.
CR095326	**When creating a new T_DOMAIN class via the MIB interface, TA_BRTHREADS cannot be set in the associated T_MACHINE class..**	
	Problem	When creating a new T_DOMAIN class via `tpadmcall()`, Tuxedo also creates an object of T_MACHINE class for the master, and allows setting the master's attributes while creating the T_DOMAIN class itself. However, when creating a new T_DOMAIN class, it currently is not possible to set TA_BRTHREADS to "Y".
	Platforms	All.
	Workaround	Create the T_DOMAIN class first, and then use the MIB to modify the T_MACHINE class entry just created.

Table 6 Known Problems for Tuxedo 8.1

CR236565	**CORBA: The generated stub/skeleton C++ files for an array of valuetype cannot be compiled.**	

	Problem	The C++ IDL compiler does not support array of valuetype definition. For example,
		`//IDL valuetype A{ ......`
		`}`
		
		`typedef A A_array[10];`
		
		The generated C++ code for the above IDL file cannot pass compilation.

	Platforms	All.
	Workaround	None.
CR238601	**CORBA valuetype: stub code for value box of some underlying boxed types does not provide enough overloaded operators or accessor/modifer functions.**	

Table 6 Known Problems for Tuxedo 8.1

	Problem	IDL compiler does not generate enough overloaded operators or accessors/modifiers functions in the C++ stub code for valuebox classes of some underlying type. These operators or functions are convenient instruments to manipulate data/members of those underlying boxed type of the valuebox class. See the following list. ■ valuebox class of sequence types: overloaded subscriptor operators are not provided. ■ valuebox class of array types: overloaded subscriptor operators are not provided. ■ valuebox class of string types: overloaded subscriptor operators are not provided. ■ valuebox class of wstring types: overloaded subscriptor operators are not provided. ■ valuebox class of struct types: underlying struct member accessor/modifier functions are not provided. ■ valuebox class of union types: underlying union member accessor/modifier functions, union discriminant functions are not provided. ■ valuebox class of any types: overloaded insertion/extraction operators of any type are not provided.
	Platforms	All.
	Workaround	First get the value/instance of the underlying boxed type using accessor method `_value()` of the valuebox class. Then use relevant overloaded operators or accessor/modifier functions of the real boxed type.
CR241622	**CORBA valuetype: valuetype with array member cannot be constructed and a core dump results.**	

Table 6 Known Problems for Tuxedo 8.1

<table>
<tr><td></td><td><b>Problem</b></td><td>If there are array type members defined in a valuetype in the IDL file, the generated C++ codes are not correct.
For example,<pre>
//IDL
typedef long LongArr[10];
valuetype A {
   ......
   public LongArr arr_val;
   ......
};
</pre>The generated C++ file may cause a runtime problem.</td></tr>
<tr><td></td><td><b>Platforms</b></td><td>All.</td></tr>
<tr><td></td><td><b>Workaround</b></td><td>None.</td></tr>
<tr><td><b>CR244880</b></td><td colspan="2"><b>Valuetype with underlying non-basic typed sequence members is not supported.</b></td></tr>
<tr><td></td><td><b>Problem</b></td><td>Valuetype definition with sequence members only support the following basic underlying types: signed/unsigned short, signed/unsigned long, octet, float, double, signed/unsigned long long, long double.
For example,<pre>
//IDL
valuetype item {
    ......
};
typedef sequence<item, 10> ItemSeq;
typedef sequence<short, 5> ShortSeq;
valuetype V {
    public ShortSeq s_val; // support
    public ItemSeq i_val; // Do not support
};
</pre></td></tr>
<tr><td></td><td><b>Platforms</b></td><td>All.</td></tr>
<tr><td></td><td><b>Workaround</b></td><td>None.</td></tr>
<tr><td><b>CR244894</b></td><td colspan="2"><b>Implementation of valuetype custom marshaling does not process wide chars correctly.</b></td></tr>
</table>

Table 6 Known Problems for Tuxedo 8.1

	Problem	CORBA specification provides custom marshaling to override the default marshaling/unmarshaling model of valuetype. Tuxedo CORBA ORB does not process wide chars correctly when custom marshaling. The `read_wchar_array()` method of `CORBA::DataInputStream` and the `write_wchar_array()` method of `CORBA::DataOutputStream` do not process wide chars properly.
	Platforms	All.
	Workaround	None.

Multibyte Character Support Limitations

The following limitations exist for the multibyte character support in BEA Tuxedo:

- UBBCONFIG and DMCONFIG parameters should not use non-ASCII characters.
- The field data on Data Dependent Routing should not use non-ASCII characters.
- The multibyte character feature is not supported in BEA Jolt.
- The multibyte character feature is not supported in BEA WebLogic Tuxedo Connector (WTC).
- The multibyte character feature is not supported in Tuxedo VIEW buffer types.
- There are no COBOL APIs equivalent to the Tuxedo APIs supporting `MBSTRING` and `FLD_MBSTRING`.

Documentation Addenda

This topic includes the following sections:

- Documentation Addenda for BEA Tuxedo 8.1
- Documentation CD-ROM Search Applet Limitations

Documentation Addenda for BEA Tuxedo 8.1

Table 7 lists documentation addenda.

Table 7 BEA Tuxedo 8.1 Documentation Addenda

CR016561	**Missing warning to avoid long blocking actions in `tpsvrinit`(3c).**	
	Problem	Documentation does not warn users to avoid long blocking actions in `tpsvrinit`(3c).
	Workaround	When invoking `tpsvrinit`(3c) in your code, avoid long blocking actions. If you do not, the following situation will occur: When one remote server in an MP configuration has trouble with `tpsvrinit()` processing, then `tmboot` fails to boot the other servers on that node. Refer to *BEA Tuxedo ATMI C Function Reference*.
CR017023	**Unclear description of the `-p` option on the `servopts`(5) page.**	
	Problem	The description of the `-p` option on the `servopts`(5) page is not explicit.
	Workaround	If the `-p` option is specified with the `L` argument, then, if the load meets or exceeds a threshold (specified by the *`high_water`* argument) for a specified amount of time (in seconds), the system will spawn additional servers. If, however, the value of *`high_water`* is 1, then the single server responsible for spawning another server will not do so as long as it is handling messages. This problem will persist as long as there is only one request waiting on the queue: the server will process it once it finishes its current request and it will not need to start a new server. However, when additional requests start arriving and waiting on the queue, then you should eventually see new servers getting started. Again, the new servers will be started when the currently running server finishes processing the current request and starts checking for the next one. Every time a server returns to its queue to get a new message to process, it checks the conditions governing the need for new servers. If those conditions are met, the server spawns exactly one new server. Refer to *File Formats, Data Descriptions, MIBs, and System Processes Reference*.
CR018788	**Documentation is incorrect on the effect of a call to `tpenqueue()` or `tpdequeue()` on the `TPQCTL` structure.**	

Table 7 BEA Tuxedo 8.1 Documentation Addenda (Continued)

	Problem	Documentation of the effect, on the TPQCTL structure, of a call to tpenqueue() or tpdequeue() includes an erroneous statement: "no value was provided when the message was queued."
	Workaround	Ignore the parenthetical statement (shown in italic): "The following is a list of valid bits for the *flags* parameter controlling output information from tpdequeue().... If a value is not available (*that is, no value was provided when the message was queued*) or the bit is not set when tpdequeue() is called, tpdequeue() completes with the flag turned off." Refer to *BEA Tuxedo ATMI C Function Reference*.
CR019383	**Missing requirements for running EventBroker servers in an MP configuration.**	
	Problem	The TMUSREVT(5) and TMSYSEVT (5) pages do not provide the requirements for running these primary EventBroker servers in an MP configuration that includes more than one release of the BEA Tuxedo system.
	Workaround	If you are setting up an MP configuration that includes more than one release of the BEA Tuxedo system and you want to run the TMUSREVT and/or TMSYSEVT server, you must run these servers on the node with the highest available release of the system. Refer to *File Formats, Data Descriptions, MIBs, and System Processes Reference*.
CR019784	**Some Windows NT and UNIX syntax errors in BEA Administration Console Online Help.**	
	Problem	The syntax for pathnames (specifically the use of slashes and backslashes) on UNIX and Windows systems is reversed in various sections of the Online Help for the BEA Administration Console.
	Workaround	Pathnames on UNIX systems are specified with slashes. Pathnames on Windows systems are specified with backslashes.
NA	**The Run Simpapp step is incorrect in the BEA Administration Console Online Help.**	
	Problem	The procedure provided for "Step 6: Run Simpapp" is incorrect in the Online Help for the BEA Administration Console.

Table 7 BEA Tuxedo 8.1 Documentation Addenda (Continued)

	Workaround	Modify this procedure as follows:
		1. Change the item number 2 in step 6 to read as follows:
		From the command shell that you used in step 1, set and export the `TUXCONFIG` environment variable as follows:
		`$ TUXCONFIG=your_simp_dir/tuxconfig`
		`$ export TUXCONFIG`
		2. Ignore item 5 in step 6.

The `Tobj::Factoryfinder::list_factories` methods return values.		
	Problem	The `Tobj::Factoryfinder::list_factories` methods as described in the *CORBA Programming Reference* does not mention that the number of factories returned by `list_factories` will be one more than the ones registered by the user. For example, if the user registered four factories then the number of factories returned by list_factories will be five. This change in behavior is because the OMG Transaction Service specification version 1.1 in section 2.1.2 specifies that the Transaction Factory is located using the FactoryFinder interface of the Life Cycle Service. Hence the Transaction factory is registered internally by the product with the FactoryFinder.
	Platforms	All.
	Workaround	None required.
CR087378	**Installer fails to install on TRU64 5.1.**	
	Problem	The Tuxedo 8.1 installer for TRU64 5.1 requires JDK 1.3.0 or higher be in the path of the user installing Tuxedo.
	Workaround	Specify the following: `export PATH=/opt/java130/bin:$PATH` where `/opt/java130/bin` contains Java executables for JDK version 1.3.0.
CR089214	**Limitation supporting multiple Tobj_Bootstrap to the same domain.**	
	Problem	The documentation does not clearly state that multiple Tobj_Bootstraps going to the same domain is not supported.
	Workaround	None required.

Table 7 BEA Tuxedo 8.1 Documentation Addenda (Continued)

CR092416	**MBSTRING is treated the same as CARRAY. Use memcpy with MBSTRING instead of strcpy.**	
	Problem	Error occurs if using `strcpy` instead of `memcpy` when you copy the string data after `tpalloc(“MBSTRING”)`.
	Workaround	Use `memcpy` with MBSTRING.
CR094270	**GW_KEEPALIVE env does not work in Tuxedo 8.1 .**	
	Problem	Documentation specifies enabling TCP Keepalive using the GW_KEEPALIVE environment variable. For BEA Tuxedo 8.1, TCP Keepalive is enabled through the DMCONFIG file.
	Workaround	If you are using TCP Keepalive with Tuxedo 7.1 and need to migrate to Tuxedo 8.1, configure this feature using DMCONFIG. For information refer to the *File Formats, Data Descriptions, MIBs, and System Processes Reference* for how to configure DMCONFIG file to use TCP Keepalive in Tuxedo 8.1.
CR108651	**References to OpenVMS in Tuxedo 8.1 documentation should be removed .**	
	Problem	Documentation references to OpenVMS in the *File Formats, Data Descriptions, MIBs, and System Processes Reference* reference an unsupported platform. This reference and the reference to AS400 should be removed as they are not supported for Tuxedo 8.1.
	Workaround	None required.

Multithreaded CORBA C++ Client Considerations

Table 8 lists a multithreaded CORBA C++ client consideration.

Table 8 Multithreaded CORBA C++ Client Considerations

NA	Multithreaded CORBA C++ clients are supported.
	BEA Tuxedo provides support for multithreaded CORBA client applications, for both the thread-per-request and the thread-per-object concurrency strategies. Build a multithreaded CORBA client as you would any CORBA client application. Whether the client application functions as a multithreaded client application depends on the environment in which it is run. Note the following considerations for multithreaded CORBA client applications running in the BEA Tuxedo environment: - Multithreaded CORBA client applications using IIOP are supported. - Multithreaded native CORBA client applications are not supported. - Multithreaded joint client/servers are not supported. - A multithreaded CORBA client application is limited to a singe Bootstrap object. - A multithreaded CORBA client application is limited to a single logon to the BEA Tuxedo domain. - CORBA client applications that use static invocation can use multiple threads. - CORBA client applications that use the dynamic invocation interface (DII) cannot be multithreaded.

Documentation CD-ROM Search Applet Limitations

The BEA Tuxedo 8.1 documentation CD-ROM includes a standalone Java search applet to help you find topics. This section describes current limitations with the search applet.

Microsoft Internet Explorer 5.0 Bug and Patch

A software bug in Microsoft Internet Explorer 5.0 can affect applets, including the BEA Tuxedo documentation search applet. This is a known Microsoft problem and there is a Microsoft patch available. This same bug might affect late builds of the Microsoft Internet Explorer 4.x browser.

There are several possible error messages that you might encounter, including "`Error occurred`" or "`No Matches Found`," plus the following message in the Java log:

```
java.io.IOException: InternetReadFile at
com/ms/net/wininet/WininetInputStream.read...
```

To run the search applet from a Web site in Internet Explorer 5.0 you need Microsoft's Java Build 3181 (released 5/26/99). The Microsoft Java update can be downloaded from: `http://www.microsoft.com/java/download/32updates.htm`.

UNC Pathnames

The Java search applet on the BEA Tuxedo 8.1 Online Documentation CD-ROM uses precompiled search databases of topics. You must adhere to one of the following options to use the Documentation CD-ROM search feature:

- Use the CD-ROM on a local CD-ROM reader.
- Copy the content of the CD-ROM to a local drive on your system.
- Map a network drive to a remote, shared device that contains the CD-ROM or a copy of the CD-ROMs contents; in your browser, use the network drive to find and open the `index.htm` file in the `\doc\tuxedo\tux81\` directory.
- Copy the CD-ROMs content to a Web server on your corporate intranet. Make sure that `index.htm` is the default filename used by the Web server software. The `\doc\tuxedo\tux81\` directory of the BEA Tuxedo 8.1 Documentation CD-ROM contains an `index.htm` file; it is the documentation home page. If your Web server software does not allow you to use a file named `index.htm`, make a copy of `index.htm` and rename the copy to the default filename you must use, such as `default.htm`; keep both the `index.htm` file and the copied file in the same directory.

You cannot use the search applet if you accessed the CD-ROM or a copy of its content through a Universal Naming Convention (UNC) path. For example, UNC paths are used by the Windows NT Network Neighborhood. The search applet will not interpret the relative paths to the matched, target `*.htm` pages because the UNC path is added to the beginning of each link. Use one of the four recommended methods described in the preceding list.

You can use UNC paths, such as accessing the documentation CD-ROM on a Network Neighborhood system's shared CD-ROM device, for all other relative hyperlinks on the CD-ROM. Only the search applet's results list is affected by this UNC limitation.

Browser Version and Platform Limitations for the Search Applet

The Documentation CD-ROM search applet tested well on:

- Microsoft Windows 2000 systems running Netscape 4.x and Microsoft Internet Explorer 4.x and 5.x.
- Microsoft Windows 98 systems running Netscape 4.x and Microsoft Internet Explorer 4.x.
- Sun Solaris systems running Netscape 4.x.

CLASSPATH Environment Variable

On some UNIX platforms, you may encounter a browser error when you open the Online Documentation CD-ROMs Search page. For example:

```
Unable to start a java applet: Can't find 'java40.jar' in your CLASSPATH.
Read the release notes and install java40.jar properly before restarting.
```

If the search applet does not work on your UNIX platform, try using the latest Netscape browser for the platform and add the Netscape Java ARchive (JAR) file to your `CLASSPATH` environment variable. The path to the JAR file is in the directory in which you installed Netscape.

For example:

```
CLASSPATH=mytools/netscape/communicator/program/java/classes/java40.jar
```

After you revise the `CLASSPATH`, exit Netscape and then restart Netscape in the updated environment. When you access the Search page, the search feature should work properly.

Online Help Addenda

BEA Application Builder Online Help Browser Requirements

The BEA Application Builder requires Netscape Navigator, instead of Microsoft Internet Explorer. If you attempt to invoke the online help from within BEA Application Builder, and Netscape Navigator is not present on your machine, the invocation fails.

Note: For a browser that includes support for Java components, we recommend Netscape Communicator 4.61 or later.

How to Obtain Patches

To obtain patches, access the BEA Customer Support page at `http://www.bea.com/support/` and open a New Case to request the patches. BEA customers can gain access to support information by accessing the Customer Support page and registering for a Web account.

You can also contact Customer Support by using the contact information provided on the BEA Tuxedo 8.1 Customer Support Card, which is included in the product box.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages