



# BEA Tuxedo®

## Known and Resolved Issues

# Copyright

Copyright © 1995-2006 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc.

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

## Trademarks and Service Marks

Copyright © 1995-2006 BEA Systems, Inc. All Rights Reserved. BEA, BEA JRockit, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Interaction, BEA AquaLogic Interaction Analytics, BEA AquaLogic Interaction Collaboration, BEA AquaLogic Interaction Content Services, BEA AquaLogic Interaction Data Services, BEA AquaLogic Interaction Integration Services, BEA AquaLogic Interaction Process, BEA AquaLogic Interaction Publisher, BEA AquaLogic Interaction Studio, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Kodo, BEA Liquid Data for WebLogic, BEA Manager, BEA MessageQ, BEA SALT, BEA Service Architecture Leveraging Tuxedo, BEA WebLogic Commerce Server, BEA WebLogic Communications Platform, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic Log Central, BEA WebLogic Mobility Server, BEA WebLogic Network Gatekeeper, BEA WebLogic Personalization Server, BEA WebLogic Personal Messaging API, BEA WebLogic Platform, BEA WebLogic Portlets for Groupware Integration, BEA WebLogic Real Time, BEA WebLogic RFID Compliance Express, BEA WebLogic RFID Edge Server, BEA WebLogic RFID Enterprise Server, BEA WebLogic Server Process Edition, BEA WebLogic SIP Server, BEA WebLogic WorkGroup Edition, BEA Workshop for WebLogic Platform, BEA Workshop JSP, BEA Workshop JSP Editor, BEA Workshop Struts, BEA Workshop Studio, Dev2Dev, Liquid Computing, and Think Liquid are trademarks of BEA Systems, Inc. Accelerated Knowledge Transfer, AKT, BEA Mission Critical Support, BEA Mission Critical Support Continuum, and BEA SOA Self Assessment are service marks of BEA Systems, Inc.

All other names and marks are property of their respective owners.

# Contents

Known Issues . . . . .	1
Product Constraints . . . . .	19
Resolved Issues . . . . .	26



# Known and Resolved Issues

This document provides a list of known and resolved issues for Tuxedo.

- [Known Issues](#)
- [Product Constraints](#)
- [Resolved Issues](#)

## Known Issues

This document provides a list of known issues for this Tuxedo release.

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 1](#) lists known problems for BEA Tuxedo.

Table 1 Known Problems for Tuxedo

<b>1. CR012652</b>	<b>Sequences and arrays contained by a CORBA : :Any cannot be passed remotely due to their lack of a repository ID.</b>
<b>Problem</b>	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 <i>The Common Object Request Broker: Architecture and Specification</i> , 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.
<b>Platforms</b>	All.
<b>Application Type</b>	CORBA C++.
<b>Workaround</b>	Sequences and arrays can be used within other data types, such as structures, or they can be used directly as parameters.
<b>2. CR012697</b>	<b>When you use VIEWS, strings are null terminated and characters truncated.</b>
<b>Problem</b>	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.
<b>Platforms</b>	All
<b>Workaround</b>	Accommodate a null in the definition of the string.
<b>3. CR014128</b>	<b>ISL will not start if there is an underscore ( _ ) in the IP name.</b>
<b>Problem</b>	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>
<b>Platform</b>	Microsoft Windows.
<b>Workaround</b>	Ensure that the ISL is configured correctly with host and port values.
<b>4. CR016275</b>	<b>Cannot have two data types for the same field name in input and output VIEWS.</b>

Table 1 Known Problems for Tuxedo

<b>Problem</b>	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.
<b>Platforms</b>	All
<b>Workaround</b>	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. <pre>svc.setInt["INVIEW.birthdate",19980308]; svc.call[null]; String bdate = svc.getStringDef["OUTVIEW.birthdate",null]</pre>
<b>5. CR017633</b>	<b>User's jrepository file needs to be converted to use the services in JREPSVR.</b>
<b>Problem</b>	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.
<b>Platforms</b>	All
<b>Workaround</b>	BEA Jolt 1.1 user should update the BEA Jolt Repository file (jrepository) to use the services in JREPSVR.
<b>6. CR017675</b>	<b>Changing FML data types does not update JSH with changes.</b>
<b>Problem</b>	It does not appear that after changing the FML field data type from int to string the JSH gets updates.
<b>Platforms</b>	All
<b>Workaround</b>	After you change the FML field definitions, reboot JSL/JSH.
<b>7. CR018177</b>	<b>client_response() or target_response() method.</b>
<b>Problem</b>	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.

Table 1 Known Problems for Tuxedo

	<b>Platforms</b>	All.
	<b>Workaround</b>	None.
<b>8. CR018211</b>	<b>RLI administration restrictions.</b>	
	<b>Problem</b>	<p>The RLI administrative functions of registering, unregistering, and changing interceptor order are not supported while the BEA Tuxedo application is running.</p> <p>The RLI administrative functions must only be done before issuing the <code>tmbboot</code> command or after issuing the <code>tmsshutdown</code> command. BEA Tuxedo application results are not predictable if interceptors are administered while the application is running.</p>
	<b>Platforms</b>	All.
	<b>Workaround</b>	Do not register, unregister, or change the interceptor order while the BEA Tuxedo application is running.
<b>9. CR018476</b>	<b>Missing <code>host:port</code> on ISL <code>-n</code> option causes Application and TMNTS servers to crash.</b>	
	<b>Problem</b>	If you define an ISL without a <code>-n host:port</code> entry in a <code>UBBCONFIG</code> file, an application server that uses the notification server (TMNTS) and TMNTS crashes.
	<b>Platforms</b>	All.



Table 1 Known Problems for Tuxedo

<b>Workaround</b>	<p>According to RFC 810, the document that specifies the syntax for IP hostnames, hostnames cannot contain an underscore ( <code>_</code> ) character. Therefore, a listening address such as <code>//my_computer.com:8000</code> is considered invalid and processes such as <code>ISL</code> and <code>tlisten</code> fail to start.</p> <p>This is a problem on Microsoft Windows NT systems because hostnames with underscore characters are often used.</p> <p>If you have a Windows NT system whose hostname contains an underscore, do one of the following:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Add a DNS alias (without an underscore) to the machine by adding a CNAME record.</li> <li>• Add an alias to the <code>%WINDIR%\System32\drivers\etc\hosts</code> file:  <pre>123.45.67.89 my_computer.com mycomputer.com</pre> where the first entry is the IP address of the machine.</li> </ul> <p>Then perform the following procedure:</p> <ol style="list-style-type: none"> <li>1. Reboot the machine.</li> <li>2. In the <code>UBBCONFIG</code> file, use <code>my_computer.com</code> when specifying the <code>PMID</code> entry for this machine in the <code>MACHINES</code> section.</li> <li>3. In all other places in the <code>UBBCONFIG</code> file, such as the <code>NETWORK</code> or <code>SERVERS</code> sections, or when starting listening processes, such as <code>tlisten</code> on the command line, use <code>mycomputer.com</code> as the hostname.</li> </ol>
<b>10. CR019447</b>	<b>Signatures get dropped when floating point values are used.</b>

Table 1 Known Problems for Tuxedo

<b>Problem</b>	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	<p>If you are using buffer types that are not supplied by the BEA Tuxedo system, replace the <code>_tmencdec</code> function in the types switch to use an encoded format with as much precision as the native format (everywhere).</p>
<p><b>11. CR019611</b> <code>tpcall ()</code> fails when it is invoked (with the <b>TPNOCHANGE</b> flag set and FML or FML32 buffers) by clients on pre-Release 7.1 nodes.</p>	
<b>Problem</b>	<p>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 <code>tpcall ()</code> for a service on a release 7.1 node. If <code>tpcall ()</code> sends an FML or FML32 buffer and the <b>TPNOCHANGE</b> flag is set, the call fails with <code>tperrno</code> set to <code>TPEOTYPE</code>.</p> <p>This problem does not occur for Workstation clients on pre-release 7.1 nodes.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	<p>Do not set the <b>TPNOCHANGE</b> flag in <code>tpcall ()</code> when pre-release 7.1 native clients are calling, with FML buffers, services on a release 7.1 node.</p>
<p><b>12. CR020175</b> Automatic calls to <code>tpterm ()</code> may not work for multithreaded and multicontexted clients.</p>	

Table 1 Known Problems for Tuxedo

	<b>Problem</b>	The BEA Tuxedo system automatically calls <code>tpterm()</code> if a client exits without calling it explicitly. In a multicontexted and multithreaded environment, however, release 7.1 and later systems cannot detect the number of contexts that are active when the client terminates; only a single call to <code>tpterm()</code> is made. As a result, only one context is shut down.  Other limitations, especially when unsolicited thread notification is used, may also exist.
	<b>Platforms</b>	All.
	<b>Workaround</b>	Make sure that before a multicontexted client exits, it explicitly calls <code>tpterm()</code> for all active contexts.
<b>13. CR020514</b>	<b>There is a 1K limit on authentication tokens on GWTDOMAIN links.</b>	
	<b>Problem</b>	The amount of data that can be sent in authorization tokens over GWTDOMAIN links may not exceed 1024 bytes for security reasons.
	<b>Platforms</b>	All.
	<b>Workaround</b>	There is no workaround for a security provider that needs tokens larger than 1024. A security provider may be able to compress tokens.
<b>14. CR025411</b>	<b>SET operation fails to reset the state of the class from ACTIVE to OPEN.</b>	
	<b>Problem</b>	The SET operation is as follows: <pre> TA_OPERATION      SET TA_CLASS          T_APPQSPACE TA_APPQSPACENAME MYQSPACE1 TA_QMCONFIG       D:\temp\tmp.1\files/QUE1 TA_LMID L1 TA_STATE          OPE </pre> The output does not show the <code>TA_STATE</code> as <code>OPEN</code> , instead it shows it as <code>ACTIVE</code> only.  <b>Note:</b> This happens for <code>tpadmcall</code> only. The <code>SET</code> operation passes for <code>tpcall</code> , <code>tpacall</code> and <code>tpenque</code> .
	<b>Platforms</b>	Microsoft Windows 2003 Server.
	<b>Workaround</b>	Use <code>tpcall</code> or <code>tpacall</code> instead of <code>tpadmcall</code> .
<b>15. CR031385</b>	<b>Value of <code>TA_CURDISPATCHTHREADS</code> field in <code>TA_SERVER</code> class in not updated properly.</b>	

Table 1 Known Problems for Tuxedo

<b>Problem</b>	The value of the TA_CURDISPATCHTHREADS field in the TA_SERVER class is not updated properly on Windows 2003 whenever there is a new thread started for the server.
<b>Platforms</b>	Microsoft Windows 2003 Server.
<b>Workaround</b>	None.
<b>16. CR031816</b>	<b>Requests on methods using multithreading appear to time out prematurely.</b>
<b>Problem</b>	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.
<b>Platforms</b>	All.
<b>Workaround</b>	Increase the UBBCONFIG file settings for BLOCKTIME and SCANUNIT. For example: SCANUNIT 5, BLOCKTIME 6000.
<b>17. CR039550</b>	<b>Memory leak using poll_next_response() for native client.</b>
<b>Problem</b>	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.
<b>Platforms</b>	All.
<b>Workaround</b>	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.
<b>18. CR042034</b>	<b>ud32 crashes when SYSTEM_ACCESS=PROTECTED and with -C tpsysadm.</b>
<b>Problem</b>	The ud32 -C tpsysadm crashes when SYSTEM_ACCESS in the configuration file is set to PROTECTED.
<b>Platforms</b>	All.

Table 1 Known Problems for Tuxedo

	<b>Workaround</b> None.
<b>19. CR042294</b>	<b>ACL Security Failure on tpconnect does not return disconnect to OSI-TP Gateway.</b>
<b>Problem</b>	Whenever a <code>tpconnect</code> (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 <code>nw_disconnect</code> is not returned to the OSI TP 4.0 Gateway. The following error messages are output to the ULOG: <pre>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</pre>
<b>Platforms</b>	All.
<b>Workaround</b>	If it is a valid client, create the ACL entry.
<b>20. CR045148</b>	<b>Tlisten has a typo on its usage.</b>
<b>Problem</b>	The usage of <code>tlisten</code> shows as: <pre>tlisten -l naddr [-d dev] [ -u {uid-#   uid-name}] [-z {0 40 56 128}] [-Z {0 40 56 128}]</pre> <p><code>-l naddr</code> should be <code>-l nlsaddr</code> as it appears in the document. This typo misleads the user to enter in an incorrect input.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	Use this usage instead: <pre>tlisten -l nlsaddr [-d dev] [ -u {uid-#   uid-name}] [-z {0 40 56 128}] [-Z {0 40 56 128}]</pre>
<b>21. CR045947</b>	<b>NO_MEMORY exception on HP.</b>

Table 1 Known Problems for Tuxedo

<b>Problem</b>	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.
<b>Platforms</b>	HP-UX
<b>Workaround</b>	None
<b>22. CR046254</b>	<b>TA_DMCODEPAGE returns garbage.</b>
<b>Problem</b>	A DMIB test case tries to create a remote service entry using ud32; its TA_DMCODEPAGE was <i>not</i> 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.
<b>Platforms</b>	All.
<b>Workaround</b>	Ignore the output for TA_DMCODEPAGE in the return packet.
<b>23. CR050552</b>	<b>SSL URL syntax parsing not working correctly.</b>
<b>Problem</b>	A URL address list containing randomized address groups followed by comma-separated lists of addresses is not being parsed correctly.
<b>Platforms</b>	All.
<b>Workaround</b>	Do not mix URL types in a list.
<b>24. CR051108</b>	<b>tlisten.pw file not the same between BEA WebLogic Enterprise 5.1 and BEA Tuxedo 8.1 or 9.0.</b>
<b>Problem</b>	The creation of the tlisten.pw is different between BEA WebLogic Enterprise 5.1 and BEA Tuxedo 8.1 or later. The impact to the customer is in a mixed environment when there are BEA Tuxedo 8.1 or later 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.
	<b>Note:</b> 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.
<b>Platforms</b>	All.

Table 1 Known Problems for Tuxedo

	<b>Workaround</b>	To resolve this problem or prevent it from happening, a single version of the <code>\${TUXDIR}/udataobj/tlisten.pw</code> file should be copied to all systems in the domain(s). This file can either be taken from the BEA Tuxedo 8.1 or later home directory or the previous version's home directory.
<b>25. CR071576</b>	<b>tlisten.pw file not the same between WLE 5.1 and Tuxedo 8.1 or later</b>	
	<b>Problem</b>	In a mixed environment with Tuxedo 8.1 or later 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.
	<b>Platforms</b>	All.
	<b>Workaround</b>	Copy a single version of the <code>\$TUXDIR/udataobj/tlisten.pw</code> file to all systems in the domain. You can use this file from the Tuxedo 8.1 or later 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.
<b>26. CR071624</b>	<b>FML function Fgetalloc and CFgetalloc gives GPF in Windows 2003.</b>	
	<b>Problem</b>	Fgetalloc gives GPF when proper FBFR*, fieldid is passed.
	<b>Platforms</b>	Windows 2003 Server
	<b>Workaround</b>	The problem can be fixed in one of the following ways in <code>fml_nt.mak</code> : <ul style="list-style-type: none"> <li>• Use <code>buildclient</code> directly to compile from <code>.c</code> files into <code>.exe</code> files, setting <code>CFLAGS</code> (not <code>CCFLAGS</code>) to specify any extra compiler options.</li> <li>• Add the <code>/MD</code> flag to the <code>cl -c</code> line.</li> </ul>
<b>27. CR071960</b>	<b>buildclient and buildserver fails with unresolved references.</b>	
	<b>Problem</b>	<code>buildclient</code> and <code>buildserver</code> fails with unresolved references.
	<b>Platforms</b>	All
	<b>Workaround</b>	When building a client or server with the XML library ( <code>libtxml</code> ), export <code>CC=</code> the C++ compiler before using the <code>buildclient</code> or <code>buildserver</code> command.
<b>28. CR089272, CR217048</b>	<b>Jolt 8.1 and 9.0 do not work with WLS 6.1 SP4 when security context is enabled.</b>	

**Table 1 Known Problems for Tuxedo**

<b>Problem</b>	An exception is sent when a client request to WebLogic Server 6.1 (when jolt pool is configured) with Jolt 8.1 or 9.0 and the security context is enabled.
<b>Platforms</b>	All.
<b>Workaround</b>	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 or 9.0 clients cannot be used with WebLogic Server releases earlier than 7.0.
<b>29. CR092413</b>	<b>Inconsistent number of retries to connect to remote domain.</b>
<b>Problem</b>	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.
<b>Platforms</b>	All.
<b>Workaround</b>	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.”
<b>30. CR092441</b>	<b>On a 64-bit platform running Tuxedo 6.5, tppost of a VIEW or VIEW32 to Tuxedo 8.1 or 9.0 sometimes fails with LIBTUX_CAT:1555.</b>
<b>Problem</b>	On a 64-bit platform running Tuxedo 6.5, tppost () of a VIEW or VIEW32 buffer to a Tuxedo 8.1 or later node sometimes fails with:  LIBTUX_CAT:1555: ERROR: Unsolicited message encoding/decoding failed (_tmencdec(TMDECODE) tperrno=12) .
<b>Platforms</b>	All.



Table 1 Known Problems for Tuxedo

	<b>Workaround</b>	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.
<b>31. CR092647</b>	<b>tmboot fails in MP mode configuration between Tuxedo 8.1 or later and WLE 5.1.</b>	
	<b>Problem</b>	tmboot with Master as Tuxedo 8.1 or later and WLE 5.1 as non-Master fails with error message in ULOG.
	<b>Platforms</b>	All
	<b>Workaround</b>	Apply the WLE 5.1 RP60 or greater to resolve this issue.
<b>32. CR092705</b>	<b>tpcall from Tuxedo 8.1 or 9.0 to Tuxedo 7.1 fails with TPESYSTEM error in Domain configuration.</b>	
	<b>Problem</b>	If connection policy is INCOMING_ONLY in Tuxedo 8.1 or 9.0 and ON_DEMAND on Tuxedo 7.1, all the services are shown in suspend mode, and client call fails TPENOENT error.
	<b>Platforms</b>	All.
	<b>Workaround</b>	This is a problem with base Tuxedo 7.1 release. You must apply Tuxedo 7.1 RP 135 or later to resolve this issue.
<b>33. CR092867</b>	<b>All enqueue fails after Unprivileged user enqueues between WLE 5.1 and Tuxedo 8.1 or later, when security is set to MANDATORY_ACL.</b>	
	<b>Problem</b>	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 and later.
	<b>Platforms</b>	All.
	<b>Workaround</b>	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 or 9.0.
<b>34. CR216274</b>	<b>The performance of GWTDOMAIN will downgrade dramatically if the two domains located at one AIX machine.</b>	

**Table 1 Known Problems for Tuxedo**

	<b>Problem</b>	This issue has somewhat of a relationship with the ngale algorithm when the two end points of TCP located at one AIX box. Under such situation, the communication just involves system buffer transitions, and contest with IPC operations. Nagle algorithm results in performance downgrade dramatically.
	<b>Platforms</b>	AIX
	<b>Workaround</b>	When you turned off the ngale in TCP, the behavior will be normal. Tuxedo GPNET provides an environment variable called SETTCPNODELAY to control the nagle algorithm usage. Set SETTCPNODELAY to any value before GWTDOMAIN startup.
<b>35. CR220004</b>	<b>Interop: Server group cannot be migrated from Tuxedo 9.0 to Tuxedo 6.5 via MIB request.</b>	
	<b>Problem</b>	The server group cannot be migrated from Tuxedo 9.0 to Tuxedo 6.5 via MIB request. For example, ud32 accepts a MIB request input file as: <pre>TA_OPERATION      SET TA_CLASS          T_GROUP TA_STATE          MIGRATING TA_SRVGRP        GROUP1</pre> The reason is an operation code of interoperability was not handled well after Tuxedo 6.5.
	<b>Platforms</b>	All.
	<b>Workaround</b>	Use tmadmin to do the group migration. For example, with the following tmadmin commands: <pre>&gt;shutdown -R -g GROUP1 &gt;migg GROUP1</pre>
<b>36. CR222951</b>	<b>Tuxedo 9.0 may consume more memory</b>	
	<b>Problem</b>	Due to the newly added features of Tuxedo 9.0, adding plenty of new code and data into underlying Tuxedo shared libraries (e.g. libengine.so), some of them now have much bigger size than they previously did in Tuxedo 8.1. The size increase of shared libraries cause running Tuxedo system and application processes depending on those libraries to consume more memory than before.
	<b>Platforms</b>	All.

Table 1 Known Problems for Tuxedo

	<b>Workaround</b>	Add memory capacity if upgrading an existing application. See <a href="#">Installation Upgrade Considerations</a> section of Release Note.
<b>37. CR223832</b>	<b>cnsbind utility core dumps on AIX platform. Although it core dumps, it still can finish its work.</b>	
	<b>Problem</b>	This occurs when exiting and when enabling CORBA trace.
	<b>Platforms</b>	AIX.
	<b>Workaround</b>	Do not use CORBA trace.
<b>38. CR218842, CR219402</b>	<b>CORBA: limitation of C++ Compiler on AIX 5.3</b>	
	<b>Problem</b>	There is a limitation on CORBA code to be compiled with C++ compiler on AIX 5.3. Type CORBA::Any or structure containing an item of this type cannot be used to declare a global or static object. Otherwise, the execution will crash when exiting due to referring to already-destroyed TypeCode.
	<b>Platforms</b>	AIX
	<b>Workaround</b>	Use these types to declare dynamic objects.
<b>39. CR217411</b>	<b>SNMP: SMUX agent does not start on Windows.</b>	
	<b>Problem</b>	Start <code>snmp_integrator.exe</code> as a service with start parameters <code>-p</code> and <code>-r</code> . Start <code>tux_snmpd.exe</code> as a service with parameters <code>-r</code> . It exits with error "Could not start the <agentname> service on local computer".
	<b>Platforms</b>	Windows
	<b>Workaround</b>	Configure the correct TUXDIR as the system environment before starting up the SNMP agent.
<b>40. CR195891</b>	<b>Outbound IIOP connection cannot end by releasing the reference to the remote object.</b>	
	<b>Problem</b>	After the outbound user limit (set by <code>-o</code> and <code>-u</code> parameters of ISL CLOPT) being reached, the server tries to release a user by assigning an empty pointer to the reference of the remote object. In 8.0RP292, assigning 0 to the reference can release a user. In Tuxedo 8.1 and 9.0 it cannot.
	<b>Platforms</b>	Windows
	<b>Workaround</b>	None

Table 1 Known Problems for Tuxedo

<b>41. CR236565</b>	<b>CORBA: The generated stub/skeleton C++ files for an array of valuetype cannot be compiled.</b>
<b>Problem</b>	<p>The C++ IDL compiler does not support array of valuetype definition. For example,</p> <pre data-bbox="470 477 813 616"> //IDL valuetype A{ ..... }  typedef A A_array[10]; </pre> <p>The generated C++ code for the above IDL file cannot pass compilation.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	None.
<b>42. CR238601</b>	<b>CORBA valuetype: stub code for value box of some underlying boxed types does not provide enough overloaded operators or accessor/modifier functions.</b>

Table 1 Known Problems for Tuxedo

<b>Problem</b>	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.
	<p>See the following list.</p> <ul style="list-style-type: none"> <li>• valuebox class of sequence types: overloaded subscriptor operators are not provided.</li> <li>• valuebox class of array types: overloaded subscriptor operators are not provided.</li> <li>• valuebox class of string types: overloaded subscriptor operators are not provided.</li> <li>• valuebox class of wstring types: overloaded subscriptor operators are not provided.</li> <li>• valuebox class of struct types: underlying struct member accessor/modifier functions are not provided.</li> <li>• valuebox class of union types: underlying union member accessor/modifier functions, union discriminant functions are not provided.</li> <li>• valuebox class of any types: overloaded insertion/extraction operators of any type are not provided.</li> </ul>
<b>Platforms</b>	All.
<b>Workaround</b>	First get the value/instance of the underlying boxed type using accessor method <code>_value()</code> of the valuebox class. Then use relevant overloaded operators or accessor/modifier functions of the real boxed type.
<b>43. CR241622</b>	<b>CORBA valuetype: valuetype with array member cannot be constructed and a core dump results.</b>

Table 1 Known Problems for Tuxedo

<b>Problem</b>	<p>If there are array type members defined in a valuetype in the IDL file, the generated C++ codes are not correct.</p> <p>For example,</p> <pre data-bbox="471 439 827 652">//IDL typedef long LongArr[10]; valuetype A {     .....     public LongArr arr_val;     ..... };</pre> <p>The generated C++ file may cause a runtime problem.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	None.
<b>44. CR244880</b>	<b>Valuetype with underlying non-basic typed sequence members is not supported.</b>
<b>Problem</b>	<p>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.</p> <p>For example,</p> <pre data-bbox="471 1046 1065 1329">//IDL valuetype item {     ..... }; typedef sequence&lt;item, 10&gt; ItemSeq; typedef sequence&lt;short, 5&gt; ShortSeq; valuetype V {     public ShortSeq s_val; // support     public ItemSeq i_val; // Do not support };</pre>
<b>Platforms</b>	All.
<b>Workaround</b>	None.
<b>45. CR244894</b>	<b>Implementation of valuetype custom marshaling does not process wide chars correctly.</b>

Table 1 Known Problems for Tuxedo

<b>Problem</b>	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 <code>read_wchar_array()</code> method of <code>CORBA::DataInputStream</code> and the <code>write_wchar_array()</code> method of <code>CORBA::DataOutputStream</code> do not process wide chars properly.
<b>Platforms</b>	All.
<b>Workaround</b>	None.

## Product Constraints

Table 2 describes product constraints for BEA Tuxedo and provides information about recommended workarounds.

Table 2 Product Constraints

<b>1.</b>	<b>Request-level interceptors do not support Object-by-Value syntax.</b>
<b>Description</b>	<code>DataInputStream::read_Value()</code> overwrites the contents of the marshalling buffer. If a request level interceptor calls <code>DataInputStream::read_Value()</code> to decode a valuetype, subsequent attempts to decode the marshalling buffer by the ORB will fail in an unpredictable manner. To prevent this, avoid using <code>DataInputStream::read_Value()</code> in request level interceptors.
<b>Platforms</b>	All.
<b>Workaround</b>	None.
<b>2.</b>	<b>Tuxedo XDR encode/decode functions cannot correctly encode integer data types larger than 32-bit.</b>
<b>Description</b>	The XDR encode/decode functions that Tuxedo uses to pass data between different machine types currently cannot encode 64-bit integer data types containing values which would not fit in a 32-bit datatype. So if the VIEW/VIEW32 long type data is out of 32-bit range, it will be truncated during XDR encode/decode.
<b>Platforms</b>	AIX, HP, and Solaris.
<b>Workaround</b>	None.
<b>3.</b>	<b>Reserved Repository IDs.</b>

Table 2 Product Constraints (Continued)

<b>Description</b>	Use of <code>Repository</code> as an interface repository ID in a <code>#pragma ID OMG IDL</code> directive, or use of <code>pk_</code> 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.
<b>Platforms</b>	All.
<b>Workaround</b>	If you use the <code>#pragma ID OMG IDL</code> directive, do not use <code>Repository</code> or any identifier that begins with <code>pk_</code> as the repository identifier.
<b>4. Restrictions to using the Dynamic Invocation Interface (DII).</b>	
<b>Description</b>	When you use DII, you may encounter problems if you do not observe certain restrictions.
<b>Platforms</b>	All.
<b>Workarounds</b>	Adhere to the following restrictions: <ul style="list-style-type: none"> <li>• If you use <code>CORBA::Request::set_return_type()</code>, you must set the location where you want the ORB to place the result for the specified request. This can be done by using <code>CORBA::Any::replace()</code> (after setting the return type) on the <code>Any</code> reference returned by <code>CORBA::Request::return_value()</code>.</li> <li>• 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.</li> <li>• When using <code>CORBA::ORB::send_multiple_requests_deferred()</code>, 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 <code>CORBA::INV_TERM_ON_ERR</code> flag had been specified). The workaround is to not use this function; instead, call <code>CORBA::Request::send_deferred()</code> on each request in the sequence with appropriate <code>try/catch</code> statements around each call.</li> </ul>
<b>5. Remote BEA Tuxedo client logon/authenticate binary userdata cannot contain NULL's.</b>	
<b>Description</b>	With an <code>AuthType</code> level of <code>Tobj::TOBJ_APPAUTH</code> , remote BEA Tuxedo client binary <code>user_data</code> passed into the <code>logon/authenticate</code> API cannot contain NULL's.
<b>Platforms</b>	All.



Table 2 Product Constraints (Continued)

<b>Workaround</b>	It is possible for binary user_data to contain legitimate NULL's. One solution is to use the native mode rather than the IIOF mode. The native mode does not have the problem.
<b>6. Client invocation times out after 60 seconds by default.</b>	
<b>Description</b>	The default behavior for a client application invoking a BEA Tuxedo IIOF 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.
<b>Platforms</b>	All.
<b>Workaround</b>	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.
<b>7. Known problem with fonts when running the University sample applications.</b>	
<b>Description</b>	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.
<b>Platforms</b>	All.
<b>Workaround</b>	None.
<b>8. Transactions with deferred, synchronous requests experience problems.</b>	
<b>Description</b>	When using DII, if you initiate deferred synchronous requests in the context of a transaction, the transaction does not complete successfully.
<b>Platforms</b>	All.
<b>Workaround</b>	Wait until you have received responses for deferred synchronous requests before you commit the transaction. Otherwise, results are unpredictable.
<b>9. C++ reserved words in IDL for operation names.</b>	
<b>Description</b>	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.
<b>Platforms</b>	All.
<b>Workaround</b>	Change an operation name in the IDL file to a name that is not a C++ reserved word.

Table 2 Product Constraints (Continued)

<b>10. Mapping of CORBA and FML data types for applications that allow interoperability between BEA Tuxedo ATMI and BEA Tuxedo CORBA.</b>	
<b>Description</b>	<p>BEA Tuxedo applications use FML field buffers that support a limited number of data types as described in the <i>Programming BEA Tuxedo ATMI Applications Using FML</i>. 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:</p> <pre>Fadd32(fmlbuf, LongFmlFld, 0, (char*) &amp;WLE_CORBA_long, 0);</pre> <p>which you would issue prior to doing a tpcall such as:</p> <pre>tpcall((char*)SomeTuxService, (char*) fmlbuf, (char**) &amp;fmlbuf, sizeof(fmlbuf), 0)</pre> <p>Such a call will succeed on most platforms, but will fail on Compaq Tru64 UNIX.</p> <p><b>Note:</b> 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.</p>
<b>Platforms</b>	All.

Table 2 Product Constraints (Continued)

<b>Workaround</b>	<p>For BEA Tuxedo and BEA Tuxedo CORBA applications to interoperate, you must change the Fadd32 statement from:</p> <pre>Fadd32(fmlbuf, LongFmlFld, 0, (char*) &amp;WLE_CORBA_long, 0);</pre> <p>to:</p> <pre>long FML_long = WLE_CORBA_long; Fadd32(fmlbuf, LongFmlFld, 0, (char*) &amp;FML_long, 0);</pre> <p>A reverse conversion would likewise be required to receive a long value returned from a tpcall and used as a CORBA::Long.</p> <p>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:</p> <pre>CORBA::Double Teller_i::get_balance( BillingW::AccountNumber account) {     long account_l = account;         :     call_tux(m_tuxbuf, "CURRBALANCE");         : }</pre>
<b>11. The C++ IDL compiler does not generate correct codes for _copy_value() method if defined valuetype graph contains cycles.</b>	
<b>Description</b>	<p>The CORBA OMG specification is vague about how to avoid infinite loops when copying a valuetype graph that contains cycles. For example:</p> <pre>//IDL valuetype A; valuetype B {     public A foo; }; valuetype A {     public B bar; };</pre> <p>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.</p>
<b>Platforms</b>	All.

Table 2 Product Constraints (Continued)

<b>Workaround</b>	Override the <code>_copy_value()</code> 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.
<b>12. The C++ IDL compiler does not support using C++ keyword as an identifier for valuetype.</b>	
<b>Description</b>	The generated code by the C++ IDL compiler for valuetype that includes C++ keywords as an identifier is incorrect.
<b>Platforms</b>	All.
<b>Workaround</b>	Avoid using C++ keywords as the identifier for valuetype.
<b>13. The pure virtual accessor and modifier functions for valuetype private state members are not protected.</b>	
<b>Description</b>	<p>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:</p> <pre>//IDL module A {     valuetype V{         .....         private V m_v;         .....     }; };</pre> <p>// The generated code:</p> <pre>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 }</pre>
<b>Platforms</b>	All.

Table 2 Product Constraints (Continued)

<b>Workaround</b>	None.
<b>14. CORBA Object-By-Value semantics are not supported in "InProc" invokes.</b>	
<b>Description</b>	<p>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.</p> <p>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.</p> <p>For example:</p> <pre> valuetype A {     ... }; interface foo {     void op1(in A arg1); }; interface bar {     void op2(in foo arg2); }; </pre> <p>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.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	Avoid passing valuetype/valuebox in "InProc" invokes scenario, or deploy CORBA objects in different server processes.
<b>15. Valuetypes that formed circular graphs will cause a CORBA server memory leak.</b>	

**Table 2 Product Constraints (Continued)**

<b>Description</b>	<p>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.</p> <p>For examples:</p> <pre>// IDL valuetype A; valuetype B;  valuetype A {     public B foo; }  valuetype B {     public A bar; }</pre> <p>Valuetype A and B formed a circular graph.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	Avoid defining Valuetypes that will form circular graphs.
<b>16. Request-level interceptors do not support Object-by-Value syntax.</b>	
<b>Description</b>	<p>The marshalling stream is modified when request-level interceptors call <code>DataStream::read_value()</code> to decode valuetypes. The Tuxedo ORB will not re-try and consequently fails to unmarshal the input stream because the stream no longer contains a properly formatted GIOP message.</p>
<b>Platforms</b>	All.
<b>Workaround</b>	None.

## Resolved Issues

The following tables list the software and documentation problems that have been fixed in Release 9.1. Problems are listed by CR (Change Request) number.

Any software or documentation fixes that are made to Release 9.1 after the initial *BEA Tuxedo 9.1 Known and Resolved Issues* document is available will be reported in updates to this section

in the online version. The online version is available through the BEA Tuxedo 9.1 Known and Resolved Issues link at

<http://e-docs.bea.com/tuxedo/tux91/index.htm>

[Table 3](#) lists problems fixed in BEA Tuxedo 9.1.

**Table 3 Problems Fixed in BEA Tuxedo 9.1**

CR Number	Problem Description
1. CR060177	Tuxedo 7.1 - <code>tpenqueue()</code> to a remote domain occasionally takes about 1 second.
2. CR110407	Tuxedo 8.0 - Does not return -1 as ORACLE timeout is reached.
3. CR111188	Tuxedo 8.0 - GWADM Error when trying to delete dynamically added local services.
4. CR128856	Tuxedo 9.0 - Multithreaded /WS - sporadic <code>TPESYSTEM</code> received by <code>tpacall</code> with no reason.
5. CR133768	Tuxedo 8.0 - Transactional requests might cause <code>GWTDOMAIN</code> loop in <code>libgwt:gw_nw_sndrollback</code> .
6. CR193155	Tuxedo 8.1 - DBBL find the license is invalid but doesn't report the right error msg.
7. CR211085	Tuxedo 8.1 - Password is regarded as expired since unexacting time is used.
8. CR212487	<code>DeleteByTypeCode</code> doesn't register value types.
9. CR216559	Tuxedo 8.0 - In SNAX domain, del a user then add the same user will fail.
10. CR221240	<code>snmpagent</code> retrieves incorrect value for some managed object.
11. CR222038	eLink/EAM 4.1/Tuxedo 8.0 - BBL freezes during shutdown/restart GWSNAX.
12. CR222301	Tuxedo 6.5/8.0 - <code>tmadmin</code> freezes on the Slave machine in a Tuxedo 8.0/6.5 MP config.
13. CR222310	Tuxedo 6.5/8.0 - Native client gets <code>LIBTUX_CAT:1555</code> errors in conversational mode.
14. CR223012	Tuxedo 8.0 - <code>viewc32</code> temporary output file names are unacceptable to the C compiler.
15. CR226453	The CORBA application server will core dump when use <code>valuetype</code> as out parameter
16. CR227883	Tuxedo 8.0/8.1 - Problem unmarshalling <code>valuetype</code> sequences embedded within structures.
17. CR228994	Tuxedo 8.1 - Problem removing reference to nested <code>valuetype</code> .
18. CR229081	Tuxedo 8.0 - The RP340 patch produced for CR223012 is missing the <code>viewc32</code> binary.

**Table 3 Problems Fixed in BEA Tuxedo 9.1**

19. CR229515	Tuxedo 8.1 - Throwing user defined CORBA exception causes server to crash.
20. CR232012	Tuxedo 8.0 - tmshutdown hangs when DBBL is un-responsive in a MP environment.
21. CR232856	Tuxedo 8.0 - DBBL does not restart when MIB call is made after DBBL goes down.
22. CR234916	License - If <code>USERS=Unlimited</code> , Tuxed reports the number of allowed clients is 0.
23. CR235194	Tuxedo 8.1 - Can't use <code>catopen</code> for user message catalog.
24. CR236051	<code>tmadmin-&gt;printinterface</code> command doesn't show value for "Routing Name" item
25. CR237057 propagated from CR209212	Tuxedo 8.0 rp316 - Interdomain transaction <code>TRANTIME</code> superseded by <code>BLOCKTIME</code> .
26. CR237339	Tuxedo 8.1 - CORBA - <code>CONCURR_STRATEGY</code> issue.
27. CR237531	Tuxedo 9.0 - Typo on <code>LIBTUX 681</code> .
28. CR237564	Oracle RAC Support.
29. CR238038	The length of <code>groupname</code> in <code>ubb</code> should be less than 30, but e-docs says it can be equal to 30.
30. CR238060	Tuxedo 9.0 - Add-on <code>cat_id</code> numbers looked up in <code>\$TUXDIR/locale/CATNAMES</code> do not work.
31. CR238895	Tuxedo 9.0 - RP installer on unix doesn't include directory "help".
32. CR239797	Tuxedo 8.1 - <code>catopen</code> still dose not work for a non-complete pathname even if <code>TM_OS_CAT</code> is set.
33. CR240684	Segmentation fault at <code>tpxmltofml32</code> in <code>stockclient (xmlfmlapp)</code> with <code>TMTRACE</code> on Solaris.
34. CR240888	Tuxedo 8.1/SOLARIS 9 - <code>TPEOTYPE</code> on <code>tpreceive</code> when <code>data=NULL/datalen=0</code> .
35. CR241008	Tuxedo 9.0 - /Domain Failover using <code>FAILOVERSEQ</code> doesn't work on Solaris.
36. CR242064 propagated from CR205433	Tux8.1 - display C locale messages when national locale message do not exist.
37. CR245074	Tuxedo 8.1 - Japanese message (Shift-JIS) outputs garbled character.



**Table 3 Problems Fixed in BEA Tuxedo 9.1**

38. CR246080	Tuxedo 8.0/8.1/9.0 - 'core dump' by <code>tmloadcf</code> when <code>SERVERS</code> keyword missed in <code>ubb</code> .
39. CR247083	Tux81/32bit/RP217 - <code>GWTDOMAIN</code> cores with <code>CMDTUX_CAT: 819</code>
40. CR250202	Tuxedo 6.5 - Please suppress <code>LIBTUX_CAT: 1370/1367</code> when booting <code>BBL</code>
41. CR250659	Oracle RAC fix for incorrect <code>LIBTUX_CAT: 1380</code> messages.
42. CR252287	Reopens dynamic registering <code>RM</code> after <code>RM</code> failure in app server.
43. CR254378	Tuxedo 8.0 (64-bit) <code>Tmboot</code> core when <code>LD_LIBRARY_PATH</code> length exceeds 2048 bytes
44. CR257541	Tuxedo 6.5 - <code>LIBTUX_CAT216</code> outputs when restarting processes on slave site.
45. CR257941	Tuxedo 9.0 - Cannot get <code>MIB</code> information of <code>TMS</code> on Windows.
46. CR257944	Tuxedo 9.0 - Multi-threaded <code>GWTDOMAIN</code> startup fails with <code>LIBGWT_CAT: 5354</code> error message.
47. CR258014	Tuxedo 8.0 - <code>/WSC</code> fails to connect to <code>WSH</code> due to <code>LIBTUX_CAT: 6249</code> error.
48. CR259044	Tuxedo 8.0 - <code>GP_CAT: 1305: ERROR: tpsubscribe</code> failed due to <code>TPEPERM</code> - bad permissions
49. CR260210	Tuxedo 9.0 - <code>.Net</code> wrapper bug in <code>libwscdnet.dll</code> .
50. CR261674	<code>setenv.sh</code> for <code>xml2fml</code> doesn't include Tuxedo <code>DIR/bin</code> in <code>PATH</code> .
51. CR263578	The Jolt Connection Pool Monitoring page is not available.
52. CR266220	Tuxedo 9.0 - <code>.Net</code> sample <code>unsolapp</code> and <code>callapp</code> sometimes fail
53. CR267048	Tuxedo 9.0 - <code>.Net</code> <code>unsolapp</code> sample result is unpredictable.
54. CR267274	Tuxedo 9.0 - <code>.Net</code> Tux wrapper <code>C#</code> client has no access to <code>TPTAIL</code> reply data.
55. CR268214	Fix <code>GWTDOMAIN DM_PW</code> 64-bit Big Endian interop with Tux 6.5.
56. CR269757	Tuxedo 9.0 - <code>.Net expt_tp_prd: TPESVCFAILUREException</code> failed

