



THE ENTERPRISE MIDDLEWARE SOLUTION

BEA TUXEDO

Glossary

Software Version 6.5
Document Edition 6.5
February 1999

Copyright

Copyright © 1999 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

BEA and ObjectBroker are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA Manager, BEA MessageQ, Jolt and M3 are trademarks of BEA Systems, Inc. TUXEDO is a registered trademark in the United States and other countries.

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

BEA TUXEDO Glossary

Document Edition	Date	Software Version
6.5	February 1999	BEA TUXEDO 6.5

Contents

Glossary

abort.....	G-1
Access Control List (ACL).....	G-1
access machine	G-1
ACID properties	G-1
ACL	G-1
activate.....	G-2
active server.....	G-2
ADE.....	G-2
administrative API.....	G-2
administrative domain	G-2
administrator.....	G-2
AEQ.....	G-2
AET	G-2
after image	G-2
allocation	G-2
alternate facility	G-3
AP	G-3
API.....	G-3
applet	G-3
application	G-3
application context name (ACN).....	G-3
application development environment (ADE).....	G-3
application entity	G-3
application entity qualifier (AEQ).....	G-3
application entity title (AET).....	G-3
application framework.....	G-4
application program (AP).....	G-4
application programming interface (API)	G-4
application to transaction monitor interface (ATMI)	G-4
architecture	G-4

argument	G-4
assembler	G-4
assembler language	G-5
asynchronous	G-5
asynchronous process	G-5
asynchronous request	G-5
ATMI	G-5
audit trail	G-5
authentication	G-5
authorization	G-5
autoinstall	G-5
availability	G-6
backup	G-6
BBL	G-6
BEA Transaction Processing	G-6
BEA TUXEDO application	G-6
BEA TUXEDO client	G-6
BEA TUXEDO domain	G-6
BEA TUXEDO server	G-6
BEA TUXEDO system	G-7
before image	G-7
bit	G-7
bitmap	G-7
blocking	G-7
blocking mode	G-7
BRIDGE	G-7
broadcast	G-7
broker	G-7
buffer types	G-8
bulletin board	G-8
Bulletin Board Liaison (BBL)	G-8
business-event middleware	G-8
byte	G-8
C++	G-8
cache	G-8
call	G-8
CARRAY buffer	G-9
catalog	G-9
CCR	G-9
channel	G-9
class library	G-9

client	G-9
client naming	G-9
client program.....	G-9
client/server computing	G-10
closed framework	G-10
COBOL.....	G-10
command-line interface	G-10
commit.....	G-10
compact disc-read only memory (CD-ROM).....	G-10
component	G-10
concurrent	G-11
configuration.....	G-11
configuration set	G-11
configure.....	G-11
connection.....	G-11
connection-oriented communication	G-11
consistent state.....	G-11
conversation.....	G-11
conversational.....	G-11
conversational communication	G-11
conversational server	G-12
conversational service	G-12
daemon	G-12
daisy chain	G-12
DASD	G-12
database	G-12
DataBase2 (DB2).....	G-12
database management system (DBMS).....	G-12
data-dependent routing	G-12
data independence	G-13
data transfer protocol	G-13
DB2.....	G-13
DBBL	G-13
DBMS.....	G-13
DDE.....	G-13
DDE conversation	G-13
deadlock.....	G-13
decoding	G-13
default.....	G-13
deferred synchronous communication.....	G-14
deployment	G-14

design document	G-14
design pattern	G-14
dialog	G-14
Direct Access Storage Device (DASD).....	G-14
Distinguished Bulletin Board Liaison (DBBL)	G-14
distributed application	G-14
distributed application framework	G-14
distributed computing	G-15
distributed transaction.....	G-15
distributed transaction processing (DTP)	G-15
DLL	G-15
domain	G-15
dotted-decimal notation	G-15
DTP.....	G-15
dynamic argument	G-15
Dynamic Data Exchange (DDE)	G-16
dynamic datatype	G-16
dynamic link libraries (DLL).....	G-16
electronic commerce.....	G-16
encoding.....	G-16
encryption	G-16
environment variable	G-16
event.....	G-16
Event Broker/Monitor.....	G-16
event posting	G-17
event subscription	G-17
extensibility	G-17
External Data Representation (XDR)	G-17
failback	G-17
failover	G-17
field.....	G-17
field manipulation language (FML)	G-18
field table	G-18
FML	G-18
FML buffer	G-18
foreign access path.....	G-18
foreign node	G-18
format independence.....	G-18
framework.....	G-18
gateway	G-19
gateway server	G-19

global transaction.....	G-19
Global Transaction Identifier (GTRID).....	G-19
graphical user interface (GUI).....	G-19
group.....	G-19
GUI.....	G-19
handler.....	G-19
hierarchical database.....	G-20
hierarchy.....	G-20
high-level language.....	G-20
host.....	G-20
host computer.....	G-20
hypertext markup language (HTML).....	G-20
inactive server.....	G-20
information hiding.....	G-20
infrastructure.....	G-20
integration.....	G-21
interaction model.....	G-21
interactive.....	G-21
interactive interface.....	G-21
internationalization.....	G-21
International Standards Organization (ISO).....	G-21
Internet.....	G-21
Internet Protocol Address (IP Address).....	G-21
interoperability.....	G-22
intranet.....	G-22
IP Address.....	G-22
ISO.....	G-22
journaling.....	G-22
Kerberos protocol.....	G-22
Kerberos security.....	G-22
keyword.....	G-22
LAN.....	G-22
LAN partition.....	G-22
lazy connection.....	G-23
legacy application.....	G-23
listener.....	G-23
LMID.....	G-23
load balancing.....	G-23
local.....	G-23
local application names.....	G-23
Local Area Network (LAN).....	G-23

local node.....	G-23
local system	G-24
local transaction.....	G-24
location transparency	G-24
log file.....	G-24
logical machine ID (LMID).....	G-24
machine.....	G-24
makefile	G-24
managed object	G-24
Management Information Base (MIB)	G-24
mapping	G-25
master node.....	G-25
message.....	G-25
message catalog	G-25
MIB.....	G-25
middleware	G-25
migrate	G-25
model	G-26
modeling	G-26
module	G-26
MP model	G-26
Multiple Virtual Storage (MVS).....	G-26
multiprocessor	G-26
multiprogramming	G-26
multithreading.....	G-26
mutual authentication	G-26
MVS.....	G-26
MVS console	G-27
name server.....	G-27
native client.....	G-27
native node.....	G-27
network	G-27
network address	G-27
network provider.....	G-27
NLS.....	G-27
node	G-28
nonblocking mode	G-28
nonmaster node.....	G-28
nonpartitioned	G-28
n-tier client/server	G-28
octet	G-28

OLTP	G-28
online transaction processing (OLTP).....	G-28
open framework.....	G-29
open system	G-29
Open Systems Interconnect Commitment, Concurrency, and Recovery (OSI CCR)	
G-29	
Open Systems Interconnection (OSI).....	G-29
Open Systems Interconnect Transaction Processing (OSI TP)	G-29
OSI CCR.....	G-29
OSI TP	G-29
out-of-band data.....	G-29
parameter	G-29
partition.....	G-30
PID.....	G-30
PL/I.....	G-30
platform	G-30
portability	G-30
port number	G-30
precommit.....	G-30
private	G-30
private MIB	G-30
procedure	G-30
process ID (PID).....	G-31
processing element	G-31
processor.....	G-31
profile	G-31
protocol.....	G-31
provider.....	G-31
queue.....	G-31
record	G-31
recover	G-31
recovery	G-32
relational database	G-32
reliability.....	G-32
remote	G-32
remote client	G-32
remote file sharing (RFS)	G-32
remote node	G-32
remote procedure call (RPC).....	G-32
remote service name	G-32
request.....	G-33

request/reply communication	G-33
request/response communication.....	G-33
request/response server	G-33
request/response service	G-33
requester.....	G-33
requestor	G-33
resource manager (RM)	G-33
resource manager instance.....	G-34
response time	G-34
RM	G-34
roll back	G-34
rollback	G-34
RPC.....	G-34
run-time trace.....	G-34
scalability.....	G-34
scope	G-34
security.....	G-35
security principal	G-35
server	G-35
server abstraction.....	G-35
server group	G-35
server ID	G-35
service.....	G-35
service code	G-35
service request	G-35
service routine.....	G-36
SERVICES section.....	G-36
SHM model.....	G-36
single threading.....	G-36
socket	G-36
socket descriptor	G-36
socket ID.....	G-36
socket number.....	G-36
SQL.....	G-36
standard MIB	G-36
StarLAN.....	G-37
state	G-37
stateful application.....	G-37
stateless application	G-37
STRING buffer	G-37
Structured Query Language (SQL).....	G-37

subroutine	G-37
subscriber	G-37
synchronization.....	G-37
synchronous.....	G-38
synchronous communication	G-38
synchronous process	G-38
system administration.....	G-38
system administrator.....	G-38
system operations	G-38
TAGENT	G-38
task.....	G-38
TCP/IP	G-38
terminal.....	G-39
thread	G-39
threading	G-39
thread of control	G-39
three-tier client/server architecture.....	G-39
TLI.....	G-39
tlisten	G-39
TLOG	G-39
TM	G-39
TMS.....	G-39
TP	G-40
TP monitor.....	G-40
TP protocols.....	G-40
TPSUT.....	G-40
transaction.....	G-40
transaction coordinator	G-40
transaction log (TLOG)	G-40
transaction manager (TM)	G-40
Transaction Manager Server (TMS).....	G-41
transaction per second (TPS).....	G-41
transaction processing (TP)	G-41
transaction processing monitor (TP monitor).....	G-41
transaction processing protocols (TP protocols)	G-41
transaction processing service user title (TPSUT)	G-41
translation	G-41
Transmission Control Protocol/Internet Protocol (TCP/IP).....	G-41
transport interface	G-42
Transport Layer Interface (TLI)	G-42
transport provider	G-42

TUXCONFIG file.....	G-42
TUXEDO.....	G-42
TUXEDO domain.....	G-42
two-phase commit (2PC).....	G-42
two-tier client/server.....	G-42
TX interface.....	G-42
type conversion.....	G-43
typed buffer.....	G-43
UBBCONFIG file.....	G-43
uniprocessor.....	G-43
Universal Device List (UDL).....	G-43
user datagram protocol (UDP).....	G-43
view.....	G-43
VIEW buffer.....	G-43
VIEW definitions.....	G-43
virtual machine.....	G-44
virtual one-hop network.....	G-44
Volume Table of Contents (VTOC).....	G-44
VTOC.....	G-44
/WS.....	G-44
WAN.....	G-44
Web GUI.....	G-44
Wide Area Network (WAN).....	G-44
window.....	G-44
Workstation.....	G-44
Workstation handler (WSH).....	G-45
Workstation listener (WSL).....	G-45
wrap.....	G-45
XA.....	G-45
XATMI application service element.....	G-45
XATMI interface.....	G-45
X_C_TYPE buffer.....	G-45
X_COMMON buffer.....	G-45
XDR.....	G-46
X_OCTET buffer.....	G-46
X/Open.....	G-46
X/Open Distributed Transaction Processing (DTP) model.....	G-46

Glossary

abort

Terminate a transaction so that all protected resources, such as database records, have the same value they had at the beginning of the transaction.

Access Control List (ACL)

A BEA TUXEDO security feature that controls client access to services by means of lists that are automatically checked each time a service is requested.

access machine

The processor within the administrative domain of an application at which a client first accesses the system. For a native client, this is the processor on which it is running. For a workstation client, this is the site at which it contacts the application.

ACID properties

The essential characteristics of transaction processing systems include the following:

- ◆ Atomicity—All changes that a transaction makes to a database are made permanent, or else all are nullified.
- ◆ Consistency—A successful transaction transforms a database from a previous valid state to a new valid state.
- ◆ Isolation—Changes that a transaction makes to a database are not visible to other operations until the transaction completes its work.
- ◆ Durability—Changes that a transaction makes to a database survive future system or media failures.

ACL

See **Access Control List (ACL)**.

activate

To move a server from the inactive (unavailable) state to the state in which it is up and running (active).

active server

A BEA TUXEDO server that is either processing a service request or is available to do so.

ADE

See **application development environment (ADE)**

administrative API

An Application Programming Interface that uses the BEA TUXEDO ATMI for the purpose of configuring and controlling an application.

See also **application programming interface (API)**.

administrative domain

That portion of an application that is actively administered at run time by a BBL process. It does not include workstations or host processors.

administrator

The person who installs the BEA TUXEDO system, configures and monitors a BEA TUXEDO application, and updates application information (such as computer names and locations).

AEQ

See **application entity qualifier (AEQ)**.

AET

See **application entity title (AET)**.

after image

A record of the contents of a data element after it has been changed. *After images* are used for forward recovery.

allocation

The assignment of various types of programs and record categories for system storage locations, such as main storage or disk storage.

alternate facility

In distributed transaction programming, a session that a transaction obtains by means of an `ALLOCATE` command.

AP

See **application program (AP)**.

API

See **application programming interface (API)**.

applet

An interactive Java program that can be run via a Web page displayed by a Java-capable browser. The applet enhances the Web page's display or enables the user to complete some specific task.

application

One or more BEA TUXEDO domains cooperating to support a single business function. Each domain comprises servers, services, and associated resource managers defined by a configuration file (`TUXCONFIG`).

application context name (ACN)

A set of rules that govern associations between application entities.

application development environment (ADE)

A set of tools (often presented or accessed via a GUI) to help programmers build applications.

application entity

A set of software components that make up a distributed transaction processing application on one computer system.

application entity qualifier (AEQ)

The locally unique component of a title that is used to identify an OSI TP application entity.

application entity title (AET)

The globally unique component of the title that is used to identify an OSI TP application entity.

application framework

The software that provides the infrastructure that makes it possible for sets of applications or other software components to work together.

application program (AP)

A single instance of a user program that performs one or more specific tasks. An AP defines transaction boundaries and accesses resources within those boundaries; it interacts with other system components using interfaces specified in the X/Open Distributed Transaction Processing model. An AP is a single thread of control involved in at most one global transaction at any time.

application programming interface (API)

1) The verbs and environment that exist at the application level to support a particular system software product. 2) A set of code that enables a developer to initiate and complete client/server requests within an application. 3) A set of calling conventions that define how to invoke a service. A set of well-defined programming interfaces (entry points, calling parameters, and return values) by which one software program utilizes the services of another.

application to transaction monitor interface (ATMI)

Application-Transaction Monitor Interface is the application programming interface to the BEA TUXEDO system that includes transaction routines, message handling routines, service interface routines, and buffer management routines.

architecture

1) A hardware and software platform (for example, SUN4.1.1, PYRAMID, or AVIION). 2) The structure and interrelationship of components in a system or in an environment.

argument

1) An independent variable (ISO). 2) Any value of an independent variable (ISO). 3) A value that a client provides when invoking an operation.

assembler

A program that converts an assembler language source program into an object program. Before assembly, a source program must be processed by a translator to convert commands into assembler language. Before execution, an object program must be processed by a linkage editor to produce a load module.

assembler language

A source language that includes symbolic machine-language statements in which there is a one-to-one correspondence with the instruction formats and the data formats of the computer. Before execution, an assembler-language application program must be processed by the translator, assembler, and linkage editor.

asynchronous

An event that occurs at a time that is unrelated to the time at which another event occurs. The two events are mutually asynchronous. The relationship between the times at which they occur is unpredictable.

asynchronous process

A process that executes independently of another process. When a request is processed asynchronously, the client application continues to perform other operations while it waits for the service request to be filled.

asynchronous request

A request that lets the client do other work while the request is being processed, enhancing parallelism within an application.

ATMI

See application to transaction monitor interface (ATMI).

audit trail

A manual or computerized means for tracing the transactions affecting the contents of a record.

authentication

The validation process used by the server to identify an entity, such as a user or a process, before allowing that user or process to join an application. This process relies on the use of passwords and other security mechanisms.

authorization

The process of determining what an entity, such as a user or a process, is allowed to access.

autoinstall

A method of creating and installing terminal definitions dynamically as terminals log on, and deleting them at logoff.

availability

Features of transaction processing systems that contribute to the smooth, continued systems operation in the presence of failures.

backup

The act of a resource manager in using a log to restore resources to some predetermined state by applying log entries sequentially to the resource until the desired state has been achieved.

BBL

See **Bulletin Board Liaison (BBL)**.

BEA Transaction Processing

See **transaction processing (TP)**.

BEA TUXEDO application

See **application and domain**.

BEA TUXEDO client

A program that invokes operations to talk to BEA TUXEDO servers. Remote and native clients are the same. Their requests are handled differently and transparently, depending on whether the client is located on a machine that is running in the BEA TUXEDO domain. BEA TUXEDO native clients are always colocated with the BEA TUXEDO domain in the same machine.

See also **client**.

BEA TUXEDO domain

A collection of servers, services, and associated resource managers defined by a single UBBCONFIG or TUXCONFIG configuration file.

See also **domain, TUXCONFIG file, and UBBCONFIG file**.

BEA TUXEDO server

A program that performs a task requested of it by a client application.

See also **server**.

BEA TUXEDO system

A robust middleware engine from BEA Systems, Inc. for developing and deploying business-critical client/server applications. It handles distributed transaction processing, application messaging, and the full complement of services necessary to build and run enterprise-wide applications.

before image

A record of the contents of a data element before the latter is changed. *Before images* are used to back out incomplete or incorrect changes in the event of a failure.

bit

The smallest unit of computer information. It has two possible states, usually represented by the binary digits 0 and 1.

bitmap

In temporary storage, a control block used by inpartition transient data to show the VSAM control intervals that have been used and are available. It is updated whenever a control interval or track is assigned to or released from a destination.

blocking

The process of combining two or more records into one block.

blocking mode

A synchronous style of message delivery, where a program must wait for an action to complete before the program can proceed. Contrast with nonblocking mode.

BRIDGE

The BEA TUXEDO system process that maintains virtual circuits to other nodes participating in an application for the purpose of transferring application messages between the nodes.

broadcast

To send the same message to every node on a network.

broker

A system-level entity whose role is to maintain subscriptions and to cause subscribers' actions to occur when events are posted.

buffer types

An abstract name for a message type. BEA TUXEDO provides four predefined types for message communication: FML, VIEW, STRING, and CARRAY. These buffer types are transparently encoded and decoded across a network of heterogeneous machines. Applications can define additional types.

bulletin board

A collection of shared data structures designed to keep track of a running BEA TUXEDO system application. It contains information about servers, services, clients, and transactions pertaining to a BEA TUXEDO application. The bulletin board is replicated on each logical native (non-foreign) machine in the application.

Bulletin Board Liaison (BBL)

A BEA TUXEDO administrative process responsible for maintaining a copy of the bulletin board on a particular processor. When the system is running, one BBL process runs continuously on each logical machine in the application.

business-event middleware

Middleware that integrates applications by providing program-to-program connection, communication, and data transfer through a publish-and-subscribe communications paradigm.

byte

A sequence of eight adjacent bits that are operated on as a unit.

C++

An object-oriented programming language developed at AT&T Bell Laboratories in the early 1980s. C++ is a “hybrid” language based on the non-object-oriented C language.

cache

A subset of memory that contains copies of the frequently accessed parts of a larger memory.

call

An instruction in COBOL, assembler language, C/370, or PL/I format that is used by an application program to request services.

CARRAY buffer

A data structure that is an array of characters any of which can be the null character. The interpretation of the array is entirely application dependent.

catalog

See message catalog.

CCR

The Commitment, Concurrency, and Recovery OSI standard.

channel

A functional unit, controlled by a processor, that handles the transfer of data between the processor and local input/output devices.

class library

A set of client programming tools. These tools can be used in a Java program or Web page-embeddable Java applet.

client

A process that generates service requests (that are sent to servers by the BEA TUXEDO system) and receives the servers' responses (which are handled by the BEA TUXEDO system). If the client is located on a machine that is part of the relevant BEA TUXEDO domain (that is, the domain to which the target servers belong), then the client is called a "native client." If the client is located on a machine that is outside that domain, then the client is called a "remote client" or a "Workstation client." (Remote clients communicate with servers through the BEA TUXEDO Workstation component.)

See also domain and server.

client naming

A BEA TUXEDO feature that enables client programs to carry both user names and client name values.

client program

Source code for a client.

See also client.

client/server computing

A programming model in which application programs are structured as clients or servers to achieve distributed processing. A client program is an application program that requests services to be performed. A server program is an entity that dispatches service routines to satisfy requests from client programs. A service routine is an application program module that performs one or more specific functions on behalf of client programs.

Client/server computing can be configured in a 2-tier or a 3-tier structure. A 2-tier configuration consists of only the client and the server. A 3-tier configuration includes a client, a server, and an intermediate level that acts as a router or a broker.

closed framework

A software infrastructure in which developers cannot remove and replace software components easily in a “plug-and-play” fashion.

COBOL

Common business-oriented language. An English-like programming language designed for business data processing applications. A standard programming language widely used for business applications, especially on IBM mainframe systems.

command-line interface

A style of user interface that allows user interaction by entering commands at a system prompt.

commit

1) Complete a transaction so that changes are recorded and stable. Protected resources are released. 2) The declaration or process of making a transaction's updates and messages visible to other transactions. When a transaction commits, all its effects become public and durable. After commitment, the effects of a transaction cannot be reversed automatically.

compact disc-read only memory (CD-ROM)

A disk from which data is read optically by laser and on which the data cannot be modified except under special conditions.

component

Part of an application.

concurrent

Pertaining to the occurrence of two or more activities within a given interval of time. Concurrent processes can alternately use shared common resources.

configuration

The set of hardware, hardware options, software, and software setup on a computer or on a network.

configuration set

The name or the number used to reference a particular configuration in a configuration partition. Each configuration set describes the services to be used when the configuration is active.

configure

To customize hardware and software for a computer or for a network.

connection

A half-duplex communication channel between processes.

connection-oriented communication

Communication between two BEA TUXEDO system processes over a connection.

consistent state

A condition in which shared data is correct and valid.

conversation

A dialog over a connection.

conversational

The attribute of communication that is described by sending data to and receiving data from another component in an iterative fashion without return to the OLTP monitor until the whole application thought is completed; the salient feature of this form of execution is that each “receive” after the first puts the process in a state of suspension until the component being addressed responds.

conversational communication

See **conversational**.

conversational server

A server that offers services that require a connection to have a conversation with the requester. The conversation follows an application established protocol. A conversational service must conform to the startup and termination rules of BEA TUXEDO system services.

conversational service

A service routine that is invoked by means of conversational communication from a client program. When the connection is established and the service is invoked, the client and service exchange data in a manner specific to the application. When the service returns, the connection ends.

daemon

A system process that runs in the background.

daisy chain

In intercommunication, the chain of sessions that results when a system requests a resource in a remote system, and the remote system discovers that the resource is in a third system and in turn makes a remote request.

DASD

See **Direct Access Storage Device (DASD)**.

database

A collection of interrelated or independent data items stored together without redundancy to serve one or more applications.

DataBase2 (DB2)

IBM relational database.

database management system (DBMS)

A program or set of programs that let users structure and manipulate the data in the tables of a database. A DBMS ensures privacy, recovery, and integrity of data in a multi-user environment.

data-dependent routing

1) Routing that directs a request to be processed by a particular group based on the value in a data field of the message. 2) A mechanism in the BEA TUXEDO system in which a service request is mapped to a specific server group based on a value contained in a designated field in the data buffer.

data independence

The ability to request data by a high-level data-management method without concern as to how the data is stored or retrieved.

data transfer protocol

A set of rules for transforming data of a particular buffer type from one representation into another.

DB2

See **DataBase2 (DB2)**.

DBBL

See **Distinguished Bulletin Board Liaison (DBBL)**.

DBMS

See **database management system (DBMS)**.

DDE

See **Dynamic Data Exchange (DDE)**.

DDE conversation

The sending and receiving of DDE messages between a client application and a server application.

deadlock

1) Unresolved contention for the use of a resource. 2) An error condition in which processing cannot continue because each of the two elements of the process is waiting for an action or a response from the other.

decoding

The process of converting bit patterns (received from a network) into data that can be translated and converted, as required. BEA Connect TPS performs ASN.1 decoding.

default

The value assumed by a program if a value is not supplied by the user.

deferred synchronous communication

A form of asynchronous communication in which one piece of software can send a message to another piece of software, and then continue to work and retrieve the reply to the message at some later time.

deployment

The process of placing an application in a distributed environment and making the application available for use. Deployment can include such tasks as installation, configuration, and administration of various parts of the application.

design document

The document written by the system integrator that explains the overall design of the application or the framework to be built.

design pattern

A document that encapsulates, in a structured format, solutions to design problems. Design patterns are guides to good design practices.

dialog

A process of sending and receiving information.

Direct Access Storage Device (DASD)

A disk, disk drive, or group of disks or drives on an IBM machine.

Distinguished Bulletin Board Liaison (DBBL)

A BEA TUXEDO administrative process that runs on the MASTER node of the application and communicates with BBLs to coordinate updates to the bulletin board.

distributed application

An application that is separated into two or more parts (such as a client and a server) on different computers that communicate through a network.

distributed application framework

A middleware suite for building and managing client/server applications. The framework also includes products providing connectivity across multiple operating environments, development services, and management.

distributed computing

An application design and implementation strategy that separates an application into units that are executed on different computers and communicate through a network. For example, an application can be separated into three distributed units: a user interface unit, a processing unit, and a storage unit.

distributed transaction

A transaction involving multiple transaction managers. In a distributed transaction environment, a client application may send requests to several servers resulting in resource updates at multiple resource managers. To complete the transaction, the transaction manager for each participant (client, servers, and resource managers) must be polled to coordinate the commit process for each participant within its domain.

distributed transaction processing (DTP)

A form of processing in which multiple application programs update multiple resources (such as databases) in a coordinated manner. Programs and resources can reside on one or more computers access a network.

DLL

See **dynamic link libraries (DLL)**.

domain

A collection of systems involved in administratively autonomous applications. The BEA TUXEDO Domains component allows for interapplication or interdomain calls and transactions to occur. The system administrator uses the `UBBCONFIG` file to configure the BEA TUXEDO domain.

See also **application** and **UBBCONFIG file**.

dotted-decimal notation

A convention for expressing IP addresses, consisting of a series of four decimal numbers (0-255), separated by periods. Example: 123.205.23.99.

DTP

See **distributed transaction processing (DTP)**.

dynamic argument

A type of argument for which a method can allocate or extend the existing storage.

Dynamic Data Exchange (DDE)

A form of communication available on Microsoft Windows, Windows NT, Windows 95, and OS/2 platforms that allows applications to exchange information through a series of messages. Two applications sending and responding to DDE messages are said to have a DDE conversation.

dynamic datatype

A datatype whose memory size is not known when the code is compiled; a dynamic datatype's memory size is known only when the code is executed.

dynamic link libraries (DLL)

A collection of functions grouped into a load module that is dynamically linked with an executable program at run time for a Windows or OS/2 application.

electronic commerce

The practice of buying and selling goods and services over the Internet.

encoding

The process of converting translated data into bit patterns that are suitable for transmission.

encryption

A process used to scramble access codes to secure data from forgery or to prevent unauthorized access.

environment variable

A string of specific value that controls a certain attribute of an application. Environment variables are made available to the application as it begins.

event

An indication to a BEA TUXEDO system process of the occurrence of a particular state or condition, for example, disconnection, transaction request mode, connection request, and so forth.

Event Broker/Monitor

A BEA TUXEDO system component that monitors the occurrence of defined system and application events and notifies subscribers when an event is detected.

event posting

Notification by the BEA TUXEDO system (or by an application) to the Event Broker/Monitor that a defined event has occurred.

event subscription

A request made to the Event Broker/Monitor to be notified when a specific event is detected.

extensibility

The ease with which a system adapts to meet new requirements. Extensibility also includes the ability to add or change a function or data (data type, file format, database schema, or information model) without:

- ◆ Requiring changes to existing functions, data, and interfaces
- ◆ Introducing unwanted side effects, such as degradation of performance, reliability, maintainability, portability, and so on

External Data Representation (XDR)

A canonical data format defined by SUN Microsystems and used to transfer data between heterogeneous hardware nodes.

failback

Restoration of message traffic to a higher priority circuit. The BRIDGE process always tries to use the highest priority circuit defined for the node; when traffic is flowing on a lower priority circuit, whether due to circuit failure or just non-availability, the BRIDGE periodically checks higher-priority circuits to find one that is usable.

failover

Seamless transfer of message traffic to a lower priority circuit on the occasion of the failure of a higher priority circuit.

field

1) In a record, a specified area used for a particular category of data. 2) An area within a segment that is the smallest referable unit of data. 3) Any designated portion of a segment. 4) A way of addressing a single item of data in a database table. A field is also an area of a window in which data is displayed.

field manipulation language (FML)

A set of C language functions for defining and manipulating storage structures called field buffers. Cooperating processes can send and receive data in fielded buffers.

field table

A file that consists of FML field names and their identifiers. The field table enables users to refer to fields by logical names rather than by system field identifiers.

FML

See field manipulation language (FML).

FML buffer

A buffer of self-describing data items accessed through the field manipulation language API.

foreign access path

A physical connection between a native BEA TUXEDO system node and a foreign node. At least one gateway server must reside on the BEA TUXEDO node.

foreign node

A node in the network that does not have access to the configuration's bulletin board, or that cannot execute the full complement of BEA TUXEDO system software.

See also field manipulation language (FML).

format independence

The ability to send data to a device without having to be concerned with the format in which the data is displayed. The same data may appear in different formats on different devices.

framework

The software environment tailored to the needs of a specific domain. Frameworks include a collection of software components that programmers use to build applications for the domain the framework addresses. Frameworks can contain specialized APIs, services, and tools, which reduce the knowledge a user or programmer needs to accomplish a specific task.

gateway

For the BEA TUXEDO system, any communication mechanism between different environments (for example, between native and foreign nodes). A software program that allows dissimilar systems to communicate and exchange information. A gateway normally handles communication between systems and performs all necessary protocol translation such that the end applications communicate transparently.

gateway server

A server process, resident on a native BEA TUXEDO system node, that communicates with one or more foreign machines.

global transaction

1) A transaction that spans one or more resource managers comprising local transactions. The Transaction Manager name for a transaction that uses multiple servers or multiple resource manager interfaces and is coordinated as an atomic unit of work. 2) The BEA TUXEDO name for a transaction that uses multiple servers or multiple resource manager interfaces and is coordinated as an atomic unit of work.

Global Transaction Identifier (GTRID)

A data structure whose value uniquely identifies a global transaction.

graphical user interface (GUI)

A high-level interface that uses windows and menus with graphic symbols instead of typed system commands to provide an interactive environment for a user. The BEA TUXEDO Web GUI, available through the World Wide Web, enables an authorized user to configure and control a BEA TUXEDO application.

group

A collection of servers or services on a machine, often associated with a resource manager. A group is an administrative unit used for booting, shutting down, and migrating servers and services.

GUI

See graphical user interface (GUI).

handler

A request that originates on a remote computer. Handlers are registered in the local BEA TUXEDO bulletin board as client programs.

See also **Workstation handler (WSH)**.

hierarchical database

A database organized in the form of a tree structure that predetermines the access paths to data stored in the database. DL/I, IMS, and SQL/DS are hierarchical database managers.

hierarchy

In a database, a tree of segments beginning with the root and proceeding downward to dependent segment types. No segment type can be dependent on more than one other segment type.

high-level language

A programming language.

host

A computer that is attached to a network and provides services other than acting as a communication switch.

host computer

The primary or controlling computer in a data communication system.

hypertext markup language (HTML)

The language used for writing pages for the World Wide Web.

inactive server

A server that is not currently available to process requests.

information hiding

A software design technique in which a piece of code contains only the information it needs to do its job.

infrastructure

A common underlying computing base. The infrastructure is a set of components (fundamental services) that support another higher-level set of components in a given system. The higher-level components are typically more directly associated with providing the specific function of the overall system.

integration

The ability of applications to share information or to process independently by requesting services and satisfying service requests. In a well-integrated system, all of the parts have a purpose, and the parts combine effectively to achieve the purpose of the overall system.

interaction model

A description of how the clients and servers in a distributed application or application framework work with each other.

interactive

Pertaining to an application in which each entry entails a response from a system or program, as in an inquiry system or airline reservation system. An interactive system may also be conversational, implying a continuous dialog between the user and the system.

interactive interface

A system facility that controls how different users see and work with the system by means of user profiles. When signing on, the interactive interface makes available those parts of the system authorized by the profile. The interactive interface has sets of selection and data entry panels through which users communicate with the system.

internationalization

A mechanism that allows customization of a system's text messages and data formats into an application's language and format of choice.

International Standards Organization (ISO)

An international organization whose membership includes standards and research groups from various nations. ISO establishes standards for computer network communications and many other technologies.

Internet

The world's largest network, the Internet is based on the TCP/IP protocol suite and is universally accessible.

Internet Protocol Address (IP Address)

A numeric value that uniquely identifies a node in a TCP/IP network. IP addresses are usually expressed in "dotted decimal notation," a series of four decimal numbers (0-255), separated by periods. Example: 123.205.23.99.

interoperability

The ability to exchange requests between entities.

intranet

A set of internal company or group-specific networks protected by firewalls and connected by IP routers. An intranet appears to its users as a single network.

IP Address

See Internet Protocol Address (IP Address).

ISO

See International Standards Organization (ISO).

journaling

The recording of information in any journal (including the system log) for possible subsequent processing by the user. The primary purpose of journaling is to enable forward recovery of the data sets. A data set can be reconstructed by applying transactions in the journal against a previous version of the data set. Journaling can be used for any other user-defined purpose, such as auditing, accounting, or performance analysis.

Kerberos protocol

The private key authentication protocol developed as part of Project Athena at the Massachusetts Institute of Technology.

Kerberos security

The security system that provides authentication, mutual authentication, and protection against replay and sequencing attacks.

keyword

1) A symbol that identifies a parameter. 2) A part of a command operand that consists of a specific character string.

LAN

See Local Area Network (LAN).

LAN partition

The failure of a LAN connecting the machines of an application, resulting in a loss of message communication between the machines. A partitioned site is one that no longer has access to the master node.

lazy connection

A network connection that is not brought up until needed.

legacy application

An existing application that needs to be modified or wrapped so that it can gain access to a BEA TUXEDO domain.

See also domain.

listener

See Workstation listener (WSL).

LMID

See logical machine ID (LMID).

load balancing

The ability of the system to ensure maximum application throughput by automatically finding the server currently doing the smallest amount of work and then sending a request to that server's queue for processing.

local

In data communication, pertaining to devices that are accessed directly (that is, without use of a telecommunication line).

local application names

The DDE Listener uses the application name supplied in the DDE Initiate message to determine if the client is looking for a local or a remote DDE application. The syntax for a local DDE application includes only the name of the application. For example, if the client is looking for Microsoft Excel on the local computer, the application name would be EXCEL.

Local Area Network (LAN)

A high-speed network that spans a limited distance, such as a building or a cluster of buildings. LANs can be connected to wide area networks (WANs) with bridge devices.

local node

The computer that is connected to a user's workstation.

local system

In a multisystem environment, the system on which an application program is being executed. A local application can process data from databases located on either the same (local) system or another (remote) system.

local transaction

A local resource manager transaction that is active on behalf of a global transaction.

location transparency

The ability to define a resource so that its name implies no specific network address or physical location.

log file

A message file that describes events that occur during an operation. Log files are updated frequently during an operation and are useful for tracing system operations and errors.

logical machine ID (LMID)

The logical name assigned (in the configuration file) to a processing element used in a transaction manager application.

machine

See processing element.

makefile

A file, referenced by the `make` command, that tells the `make` command how to create each of the files needed to generate a complete program. The makefile contains a list of source files, object files, and dependency information.

managed object

An entity (such as a process, a piece of hardware, or system performance) that is defined in the MIB and is controlled by a management device.

Management Information Base (MIB)

A BEA TUXEDO system component that provides a complete definition of the classes and their attributes that make up the BEA TUXEDO system. The BEA TUXEDO system Management Information Base comprises a generic MIB and

specific MIBs for each major component. Configuration and administration of the BEA TUXEDO system can be done programmatically by using the /AdminAPI and the ATMI to set or change the value of an attribute.

mapping

The process of associating local values or entities with values or entities that are meaningful on remote systems.

master node

The `MASTER` node for an application as designated in the `RESOURCES` section of the configuration file. It contains the master copy of the `TUXCONFIG` binary configuration file. Administration of the running system is done from the `MASTER` node.

message

Data that contains information and/or instructions for the recipient.

See also **request**.

message catalog

With respect to internationalization, a file or storage area containing program messages, command prompts, and responses to prompts for a particular native language, territory, and codeset. **message definition block** The total body of data that comprises a message definition, such as the command name, the subsystem name, and the internal and external recommendations.

MIB

See **Management Information Base (MIB)**.

middleware

A set of services for building distributed client/server applications, such as services for locating other programs in the network, establishing communication with those programs, and passing information between applications. Middleware services can also be used to resolve disparities between different computing platforms and to provide a uniform authorization model in multivendor and multioperating system networks.

migrate

Relocate a server or group of servers from one LMID to another. Migration must be planned for and specified in the configuration file.

model

A paradigm in which details have been abstracted for the sake of simplification.

modeling

A design technique used in developing architecture, simulations, and computer systems.

module

A piece of code that contains information on a particular topic and on the topic's related interfaces. For example, code that describes a banking withdrawal operation could be stored in a module.

MP model

A BEA TUXEDO application configuration that runs on more than one physical processing element. This can mean two or more uniprocessors, one or more multiprocessors, or a combination of uniprocessors and multiprocessors.

Multiple Virtual Storage (MVS)

One of IBM's principal mainframe operating systems. MVS/XA: Extended Architecture. MVS/ESA: Enterprise Systems Architecture.

multiprocessor

A computer that has more than one processing element, each with its own private memory.

multiprogramming

The concurrent execution of two or more computer programs.

multithreading

Use of a process by several transactions.

mutual authentication

The process by which peers authenticate themselves to each other. In normal authentication, a server accepts a client's proof of identity, but does not authenticate itself to the client. In mutual authentication, the client proves its identity to the server and demands that the server prove its identity to the client before communication proceeds.

MVS

See **Multiple Virtual Storage (MVS)**.

MVS console

A terminal used by the MVS operating system and other software components to display informational and error messages for the system operator. MVS consoles can also be used by the system operator to enter commands or to inquire about and modify system status.

name server

A software component of the BEA TUXEDO system that transparently maps service names to physical addresses so that users can communicate with services by name rather than by internal identifier.

native client

See client.

native node

A machine in a BEA TUXEDO configuration that contains the full complement of BEA TUXEDO software, and that has access to the same bulletin board as all other native nodes in the configuration (that is, it is part of the administrative domain of the application).

network

1) An interconnected group of nodes. 2) The assembly of equipment through which connections are made between data stations. 3) The communication path used to communicate with a server.

network address

A unique identifier assigned to each network connection made. This identifier may either be assigned randomly by the provider or created in response to a request for a specific identifier from the process establishing the connection. The format of the network address is provider-specific.

network provider

The protocol used at the transport level and below to communicate data across a network. Network providers are typically accessed from programs through a transport interface. Examples of network providers are TCP/IP and StarLAN.

NLS

The UNIX system network listener service.

node

A point on a network. The term is also used to refer to a computer (for example, a single instantiation of the UNIX operating system) that participates in a BEA TUXEDO system application. It is possible, however, to have more than one node in a multiprocessor system.

nonblocking mode

An asynchronous style of message delivery, that does not require a program to wait for an action to complete before proceeding.

nonmaster node

Any node of a BEA TUXEDO application that is not designated to be the `MASTER` node.

nonpartitioned

A term used to identify portions of a partitioned network that continue to be able to communicate with the `DBBL` on the master node.

***n*-tier client/server**

An application development approach that distributes application logic across three or more environments: the desktop computer, one or more application servers, and a database server. The main advantage of the *n*-tier client/server model is that it extends the benefits of client/server architecture to the enterprise level. Other advantages include added manageability, scalability, security, and higher performance.

octet

1) A byte that consists of eight bits. 2) A byte composed of eight binary elements.

OLTP

See **online transaction processing (OLTP)**.

online transaction processing (OLTP)

1) A form of data processing in which users at terminals or workstations send messages to application programs, which update databases in real time. 2) The execution of units of work in an environment that appears to the user as immediate; real-time; usually having internal recoverability, history-keeping, and consistency-assurance features.

open framework

A software infrastructure in which developers can easily remove and replace software components in a “plug-and-play” fashion.

open system

A system that implements specified common standards across different computer vendors. Implementing open system standards for communication allows computers from different vendors to communicate with each other.

Open Systems Interconnect Commitment, Concurrency, and Recovery (OSI CCR)

1) This protocol is an ISO standard (ISO/IEC 9804) for services and protocols that are used to commit or roll back global transaction branches. 2) A software implementation of the ISO/IEC 9804 standard.

Open Systems Interconnection (OSI)

A consortium that facilitates communication among different types of computer systems.

Open Systems Interconnect Transaction Processing (OSI TP)

1) An ISO standard (ISO/IEC 10026-2) for services and protocols that are used to establish dialogs and pass messages between clients and service routines on different computers. 2) A software implementation of the ISO/IEC 10026-2 standard.

OSI CCR

See Open Systems Interconnect Commitment, Concurrency, and Recovery (OSI CCR).

OSI TP

See Open Systems Interconnect Transaction Processing (OSI TP).

out-of-band data

Data delivered by the BEA TUXEDO system outside the normal client/server communications channels supported by the BEA TUXEDO system.

parameter

See argument.

partition

A state in which one or more active nodes of a networked application are unable to contact other active nodes because of a problem such as a LAN failure.

PID

See process ID (PID).

PL/I

A programming language designed for use in a wide range of commercial and scientific applications.

platform

The combination of hardware, operating system, and windowing system software that supports an application.

portability

The ease with which developers can move an application from one platform to another without costly reengineering.

port number

The entity on a TCP/IP host that identifies a logical communications channel and distinguishes one connection from another. A TCP/IP server “listens” for incoming connection requests at a designated port. A TCP/IP client initiates a connection with the server by specifying the host's IP address and the server's designated port number.

precommit

The process (used by resource managers to make data recoverable) of copying data changes to stable storage.

private

A field that is not visible outside its class definition.

private MIB

A MIB that is defined under a private MIB directory.

procedure

A sequence of instructions (to a computer) for performing a particular task.

process ID (PID)

A unique number that identifies a process.

processing element

A uniprocessor or one of the processors that make up a multiprocessor.

processor

See **processing element**.

profile

A set of information about a client or a user. The profile provides information the server might require to recognize the client or the user.

protocol

1) A set of rules that govern the format and timing of messages sent and received over a communication link. TCP/IP is an example of a network protocol. 2) The set of “rules” followed by two systems to communicate and exchange information.

provider

The communications product supplying networking facilities through level 4 of the OSI communications protocol.

queue

A simple data structure for managing the time-staged delivery of requests to servers. Queued elements may be sorted in some order of priority. Clients insert items in the queue and servers remove items from the queue, as soon as possible, in batch, or periodically.

record

Input or output data as it exists outside the local or remote BEA TUXEDO regions, or on different kinds of systems.

recover

A request from a coordinator or a participant to complete an identified transaction.

recovery

In transaction systems, after a failure, the ability to restore the system to the most recently committed, and therefore consistent, state. In distributed systems, recovery may involve resynchronizing several distributed components. Once a system has been recovered, processing can resume, and transactions aborted as a result of the failure can be resubmitted.

relational database

A database accessed and organized according to the relationships between data items. Relationships are expressed by means of tables that allow the accessing of items with matching attributes. The access path is determined at the time of access. SQL is an example of a relational database manager.

reliability

The extent to which a system (or part of a system) produces the correct output on repeated trials (without unintended side effects), while meeting the performance specification.

remote

Describes a service or a computer that is available to a client over the network.

remote client

See client.

remote file sharing (RFS)

A UNIX system capability that provides access to remote files over a network.

remote node

Any computer in the network other than the computer to which the user's workstation is connected.

remote procedure call (RPC)

A local procedure call that is executed in a non-local program or address space. Enables application logic to be split between a client and a server in the way that best uses available resources.

remote service name

The name (between 1 and 16 characters long) of a service offered by a remote system that can be accessed through a remote gateway.

request

A message sent by a client that identifies an operation to be performed.

See also **client** and **server**

request/reply communication

See **request/response communication**.

request/response communication

Communication characterized by a single request matched to a single response message. (The request, for a task, is made by a client. The task is performed and the response is sent by a server.) There are synchronous and asynchronous variations of request/response communication.

request/response server

A server that offers request/response services.

request/response service

A service initiated by a request from a client. The service routine receives a single request and provides (at most) a single reply. A request/response service is handled like a procedure and has the following properties: it is executed until completion, it does not have any dialog with the requestor, and it sends back a return-value to the requester. For a requester, the execution of a request/response service can be synchronous or asynchronous.

requester

A generic term for a client or server that is acting as a client.

requestor

A process that receives messages from clients, converts these messages to a common internal form, determines the appropriate server or servers for the transaction request, and forwards the request to a server.

resource manager (RM)

An interface and associated software that provides access to a collection of information and processes; for example, a database management system. Resource managers provide transaction capabilities and permanence of actions; they are the entities accessed and controlled within a global transaction.

See also **transaction manager (TM)**.

resource manager instance

A particular instance or occurrence of a resource manager (for example, the EMPLOYEE database). There may be many occurrences or instances of the same or different resource managers within a global transaction, each managing different data. Each resource manager instance is considered to be autonomous, in full control of local access (for both local and global transactions), administration, and so forth.

response time

The elapsed time between entering an inquiry or request and receiving a response.

RM

See resource manager (RM).

roll back

Terminates a transaction such that all resources updated within a transaction revert to their original (that is, pretransactional) state.

rollback

The event that ends a transaction and nullifies or undoes all changes to resources that were specified during that transaction.

RPC

See remote procedure call (RPC).

run-time trace

A BEA TUXEDO feature that enables users to monitor application-to-application transactions and, if necessary, troubleshoot distributed applications under development or during production. Also, it allows users to pinpoint problems to any hardware, operating system, network, or application code.

scalability

The extent to which developers can apply a solution to problems of different sizes. Ideally, a solution should work well across the entire range of complexity. In practice, however, there are usually simpler solutions for problems of lower complexity.

scope

To use a class to enforce a particular use for an application.

security

The protection of information from unauthorized modification or disclosure and the protection of resources from unauthorized use.

security principal

An entity that is known to, and can be authenticated by, the security system.

server

1) A process that receives service requests from a requester and dispatches the service routine that acts on the request. 2) A software module that accepts requests from clients and other servers and returns responses. 3) A software program that provides a service in a client/server architecture.

See also **client**.

server abstraction

Applications combine their service routines with the BEA TUXEDO system `main()` in building a server process. The BEA TUXEDO system's `main()` provides server initialization and termination, and receives incoming requests and dispatches them to service routines. All of this processing is transparent to applications.

server group

See **group**.

server ID

An identifier for a single server. No two servers can operate at the same time with the same server ID.

service

1) An application routine available for request by a client in the system. 2) A module of application code that carries out a service request.

service code

The name associated with a service offered remotely through a `listen` process.

service request

A request initiated by a requester process that asks for the invocation of a service.

service routine

An application module that performs one or more specific services on behalf of clients. The structure of service routines (the mechanism by which they are called and terminated) is defined by the XATMI interface specification.

SERVICES section

A section of the configuration file that is used to define services.

SHM model

A BEA TUXEDO application that runs entirely on a single processing element.

single threading

The complete execution of a program. Processing of one transaction is completed before another transaction is started.

socket

An endpoint of communication to which a name may be bound. The socket interface is a network access method supported by the BEA TUXEDO system. The logical end point of a TCP/IP connection. An application accesses a TCP/IP connection through a socket.

socket descriptor

A TCP/IP-assigned number that uniquely identifies a socket and TCP/IP connection. An application must specify the socket descriptor in TCP/IP API calls to identify the socket/connection.

socket ID

See socket descriptor.

socket number

See socket descriptor.

SQL

See Structured Query Language (SQL).

standard MIB

A MIB developed as a standard by the Internet community. Examples of the standard MIBs are MIB I and MIB II.

StarLAN

An AT&T LAN product.

state

The situation of a conversation from the point of view of one of the participating transactions. The conversation state determines the commands that a transaction can validly issue. The state of each transaction changes dynamically in the course of conversation.

stateful application

An application that retains state information in memory after a service or an operation has been performed.

stateless application

An application that flushes state information from memory after a service or an operation has been performed.

STRING buffer

A data structure that is an array of nonnull characters terminated by the null character. It is a self-describing buffer.

Structured Query Language (SQL)

A non-procedural language for defining and accessing relational databases. SQL has become the industry standard database language.

subroutine

A sequenced set of instructions that can be used in one or more programs and at one or more points in a program. The execution of a subroutine is usually invoked by a call.

subscriber

An application program that subscribes to an event or set of events, and declares what action should take place when an event is posted.

synchronization

A coordinated commitment control process between communicating transactions that ensures that all logically related updates to recoverable resources are completed or that all are backed out.

synchronous

1) Pertaining to an event that happens, exists, or arises at precisely the same time as another event. 2) Pertaining to an operation that occurs regularly or predictably with regard to the occurrence of a specified event in another process; for example the calling of an input/output routine that receives control at a precoded location in a program.

synchronous communication

A method of transmitting data using a timing signal. With this form of communication, when one piece of software sends a message to another piece of software, the originating software must wait until the service provider completes the request before it can continue.

synchronous process

A process that cannot be executed independently of other processes. When a request is processed synchronously, the client must wait until the service provider completes the request before continuing.

system administration

1) The preparation of an instance of a system for use in a particular setting or installation. 2) Changing the system as the installation changes. This term is often used in a way that includes system operations as well.

system administrator

See administrator.

system operations

Tasks that need to be performed regularly for a system. Examples are backing up and restoring data and logs, monitoring the system for error conditions, and so forth.

TAGENT

A BEA TUXEDO system process used to extend administrative capabilities to remote nodes.

task

One execution of a transaction.

TCP/IP

See Transmission Control Protocol/Internet Protocol (TCP/IP).

terminal

1) A computer monitor. 2) A point in a system or communication network at which data can either enter or leave.

thread

A unit of execution or an execution context. A sequence of instructions being executed and the memory manipulated by those instructions.

threading

The process that allows various transactions to be executed concurrently.

thread of control

A thread of control (or a “thread”) is the software module, with its context, that is currently in control of a processor. A thread is typically a UNIX process. Its context may include a current transaction, the process's locks on shared resources, and the files the process has open. The thread concept is central to the transaction manager's coordination of resource managers. Applications call resource managers to request work, while transaction managers call resource managers to delineate global transactions.

three-tier client/server architecture

An implementation of *n*-tier client/server architecture.

TLI

See **Transport Layer Interface (TLI)**.

tlisten

A network-independent listener process that runs as a daemon process and provides remote service connections for other BEA TUXEDO system processes.

TLOG

See **transaction log (TLOG)**.

TM

See **transaction manager (TM)**.

TMS

See **Transaction Manager Server (TMS)**.

TP

See **transaction processing (TP)**.

TP monitor

See **transaction processing monitor (TP monitor)**.

TP protocols

See **transaction processing protocols (TP protocols)**.

TPSUT

See **transaction processing service user title (TPSUT)**.

transaction

1) A complete unit of work that transforms a database from one consistent state to another. In DTP, a transaction can include multiple units of work performed on one or more systems. 2) A logical construct through which applications perform work on shared resources (for example, databases). The work done on behalf of the transaction conforms to the four ACID Properties: atomicity, consistency, isolation, and durability.

transaction coordinator

A system software component that provides the infrastructure that guarantees the integrity and consistency of an operation and the data involved in a transaction.

See also **transaction manager (TM)**.

transaction log (TLOG)

The BEA TUXEDO system log that keeps track of global transactions.

transaction manager (TM)

A system software component that manages global transactions on behalf of application programs. A transaction manager coordinates commands from application programs and communication resource managers to start and complete global transactions by communicating with all resource managers that are participating in those transactions. When resource managers fail during global transactions, transaction managers help resource managers decide whether to commit or roll back pending global transactions.

See also **transaction coordinator**.

Transaction Manager Server (TMS)

A BEA TUXEDO system server process that manages the 2-phase commit protocol and recovery for global transactions.

transaction per second (TPS)

A throughput rating used in conjunction with the standard transactions defined by TPC. Often the rating implies the maximum number of transactions that can be processed by the system in which 90% of the transaction's response times are less than two seconds.

transaction processing (TP)

Transaction processing consists of a set of convenient functions that make it easy for you to write code that does the following: 1) Initialize the server application and execute startup and shutdown routines. 2) Tie the server application to BEA TUXEDO domain resources. 3) Perform housekeeping functions.

transaction processing monitor (TP monitor)

A class of products that provide a transaction execution environment on top of conventional operating systems.

transaction processing protocols (TP protocols)

A set of standard protocols by which transaction processing managers on heterogeneous systems interoperate.

transaction processing service user title (TPSUT)

A value that is used to identify OSI TP endpoints that exist within one application entity.

translation

The process of changing how intrinsic data types are represented in input data and output data (with respect to word length, byte ordering, and character encoding).

Transmission Control Protocol/Internet Protocol (TCP/IP)

An open communications protocol supported by the Transport Layer Interface (TLI) and most systems, including UNIX system platforms, personal computers, and IBM mainframes. TCP/IP can be used to exchange information between dissimilar computing platforms.

See also **Transport Layer Interface (TLI)**.

transport interface

The programming interface used to access a network provider. Transport interfaces are typically network provider independent to an extent.

Transport Layer Interface (TLI)

The standard UNIX system user-level interface to data communications features as defined by level 4 of the OSI communications protocol. It is a network access method supported by the BEA TUXEDO system.

transport provider

See **network provider**.

TUXCONFIG file

The binary version of the configuration file for a BEA TUXEDO application. This file is accessed by all BEA TUXEDO processes for all configuration information.

See also **application, client, server, UBBCONFIG file**.

TUXEDO

See **BEA TUXEDO system**.

TUXEDO domain

See **domain**.

two-phase commit (2PC)

A method of coordinating a single transaction across more than one DBMS (or other resource manager). It guarantees data integrity by ensuring that transactional updates are committed in all of the participating databases, or are fully rolled backout of all of the databases, reverting to the state prior to the start of the transaction.

two-tier client/server

An application development approach that splits an application into two parts and divides the processing between a desktop workstation and a server machine.

TX interface

The Transaction Demarcation (TX) API used by APs to call the transaction manager. APs use the TX interface to define the boundaries of global transactions and direct the completion of those transactions.

type conversion

The process of converting an application program's data buffer or record so that the data is formatted in a manner that is suitable to a target application program.

typed buffer

A buffer for message communication involving data of a specific type.

See **buffer types**.

UBBCONFIG file

The ASCII version of the configuration file for a BEA TUXEDO application. This is the file from which the `TUXCONFIG` file is generated.

See also **application, client, server, TUXCONFIG file**.

uniprocessor

A computer that has only one CPU.

See also **multiprocessor**.

Universal Device List (UDL)

A system-wide list of devices (either raw disk slices or UNIX files) on which space is allocated for `TUXCONFIG` configuration tables, BEA TUXEDO transaction logs, and, possibly, databases. Its location is specified by the `FSCONFIG` or `TUXCONFIG` environment variable.

user datagram protocol (UDP)

The TCP/IP datagram transport layer protocol.

view

In the VIEW System Manager, a window that is displayed when you click on an icon that represents a managed host.

VIEW buffer

A data structure similar to a C structure. As part of defining this buffer type, a view description file is created. It is a self-describing buffer. VIEW buffers are always accompanied by VIEW definitions.

VIEW definitions

Descriptions of data structures that are used for input and output in the BEA TUXEDO environment.

virtual machine

The functional equivalent of a computer and its associated devices that is controlled by a user at a terminal.

virtual one-hop network

A network in which all nodes can be reached from all other nodes in exactly one transmission. This implies nothing about the physical configuration of the network, which may be a ring, star, bus, or any other valid configuration as long as it appears that all nodes are fully interconnected.

Volume Table of Contents (VTOC)

A file that contains the BEA TUXEDO system and possibly database tables.

VTOC

See **Volume Table of Contents (VTOC)**.

/WS

See **Workstation**.

WAN

See **Wide Area Network (WAN)**.

Web GUI

A graphical user interface for the administration of BEA TUXEDO applications, that is accessed through a Web browser. It is provided with BEA TUXEDO Release 6.5.

Wide Area Network (WAN)

A public or private data communications system in which data is transmitted primarily over telephone lines.

window

An area of a user's screen in a graphical user interface system. A window is a mechanism used by applications for interacting with a user.

Workstation

The BEA TUXEDO feature that allows remote workstations and personal computers to participate in BEA TUXEDO applications without requiring the BEA TUXEDO resources necessary to support servers on the workstation.

Workstation handler (WSH)

A surrogate client (supplied by the BEA TUXEDO system) responsible for managing a set of workstation client connections. Workstation handlers are started dynamically by the Workstation listener. The handler may be customized if necessary. This process resides within the administrative domain of the application. Handlers are registered in the local BEA TUXEDO bulletin board as clients.

Workstation listener (WSL)

The process (supplied by the BEA TUXEDO system) responsible for acting as the single point of contact for Workstation clients. The Workstation listener also handles the distribution of workstation connections to Workstation handlers, starting new handlers as necessary. This process resides within the administrative domain of the application.

wrap

To enclose an application in a software layer to make the application available to other applications.

XA

The interface between the transaction manager and resource manager.

XATMI application service element

Software that maps primitives in the XATMI interface to the OSI TP protocol.

XATMI interface

An interface that enables application programs to use request/response communication and conversational communication during global transactions.

X_C_TYPE buffer

A non-nested C structure whose elements are any of the following C data types: int, short, long, char, float, double, character string, and octet array. X_C_TYPE is one of three buffers that are defined in the X/Open XATMI standard. It is equivalent to the BEA TUXEDO VIEW buffer.

X_COMMON buffer

A non-nested C structure whose elements are any of the following C data types: short, long, or char. X_COMMON is one of three buffers that are defined in the X/Open XATMI standard. It is equivalent to the BEA TUXEDO VIEW buffer; however, X_COMMON represents only the subset of field types that are common to both the C and COBOL languages.

XDR

See External Data Representation (XDR).

X_OCTET buffer

An array of bytes whose structure is defined by an application. X_OCTET is of one three buffers that are defined in the X/Open XATMI standard. It is equivalent to the BEA TUXEDO CARRAY buffer.

X/Open

The X/Open Company, Ltd., an international private consortium of vendors and users working to establish standards for open systems. BEA TUXEDO products are being designed to implement X/Open standards for distributed transaction processing.

X/Open Distributed Transaction Processing (DTP) model

The distributed transaction processing model specified in standards developed by the X/Open Company, Ltd. The BEA TUXEDO architecture is based on these standards. The model defines four components of a DTP system: Application programs (APs) define the transaction boundaries and perform the actions that make up the transaction (typically database updates).

Resource managers (RMs), such as database management systems, provide access to shared resources. Communication resource managers (CRMs) allow application programs to communicate with each other. Transaction managers (TMs) assign unique identifiers (XIDs) to transactions, monitor the progress of transactions, and handle transaction completion or recovery.