IBM WebSphere Message Broker Security Target

Version 2.1.2

2007-08-22

Document History

Version	Date	Summary	Author
1.0	2006-10-23	Final EAL3 ST plus changes by IBM.	SAIC / IBM
1.1	2006-12-11	Fixed inconsistencies.	Scott Chapman ,atsec
1.2	2007-01-23	Minor corrections	Martyn Honeyford, IBM Scott Chapman, atsec
1.3	2007-01-30	Fixed section 1.3. Added section 2.2.2.5. Removed FMT_MSA.3b.	Scott Chapman, atsec
1.4	2007-02-01	Changed references to Broker Manger to Broker in section 5.1.2	Jason Yong, IBM
1.5	2007-02-06	Fixed FDP_ACF.1 in table 6.	Scott Chapman, atsec
1.6	2007-03-05	Fixed several evaluator comments.	Scott Chapman, atsec
1.7	2007-03-09	Added new figure 1 and supporting text.	Scott Chapman, atsec
1.8	2007-03-12	Added audit events to FAU_GEN.1.	Scott Chapman, atsec
1.9	2007-03-15	Moved O.ADMIN_GUIDANCE to OE.ADMIN_GUIDANCE and reworded. Enhanced TOE descriptions in chapter 2. Modified FDP_ACC.2.1.	Martyn Honeyford, IBM Scott Chapman, atsec
1.9.1	2007-03-15	Fixed typo in FCO_NRO.1.3.	Scott Chapman, atsec
1.9.2	2007-05-16	Listed the allowed nodes in section 2.2.1.2. Updated FDP_ACC.2.1, FDP_ACF.1.x, & section 6.1.3. Added FIA_ATD.1 & updated section 8.2.1.9, tables 5 & 6.	Scott Chapman, atsec
1.9.3	2007-05-21	Updated A.NO_EVIL and OE.CONFIG to include message flow developers.	Scott Chapman, atsec
1.9.4	2007-06-19	Updated FDP_ACF.1.3 and section 6.1.3.	Scott Chapman, atsec
1.9.5	2007-06-27	Updated 2.2.1 (physical boundaries) adding database names and versions plus MQ version.	Scott Chapman, atsec
1.9.6	2007-07-06	Added application note to FAU_GEN.1.2. Added TLS v1.0. Updated section 2.2.1.	Scott Chapman, atsec
1.9.7	2007-07-13	Add FDP_IFC.2, FDP_IFF.1, and other supporting SFRs. Removed FCO_NRO.1. Updated FDP_ACF.1.2.	Scott Chapman, atsec
1.9.8	2007-07-16	Updated O.ACCESS, FDP_IFC.2.1, and 6.1.2 (moved 6.1.3.2 into 6.1.2).	Scott Chapman, atsec
1.9.9	2007-07-16	Updated section 2.2.2.1.	Scott Chapman, atsec
2.0	2007-07-25	Changed ESQL to ESQLCompute in 2.2.1.2. Updated A.INSTALL_REQ. Changed platform to machine in 2.2.1.1. Added ‘product’ to OE.INSTALL in section 8.1.1.1. Removed ‘class’ from FDP_ACC.2.1 text. Made SSL/TLS non-mandatory in a trusted network.	Scott Chapman, atsec
2.0.1	2007-07-27	Added “authorized developer” to FMT_MSA.3.2a.	Scott Chapman, atsec
2.1	2007-08-02	Removed FCO_NRR.1, O.NON_REPUDIATION, T.MESSAGE_DENIAL. Added FAU_GEN_(EXP).1 and	Scott Chapman, atsec

		replaced FAU_GEN.1 with this new SFR. Updated table 7.	
2.1.1	2007-08-03	Updated section 1.2 from conformant to extended.	Scott Chapman, atsec
2.1.2	2007-08-22	Modified the set of audit events in FAU_GEN_(EXP).1.1 and the audit trail location in 6.1.1.	Scott Chapman, atsec

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is WebSphere Message Broker provided by IBM. WebSphere Message Broker enables information, packaged as messages to flow between different business applications, ranging from large legacy systems through to unmanned devices such as sensors on pipelines.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7), and
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – IBM WebSphere Message Broker Security Target

ST Version – Version 2.1.2

ST Date – 2007-08-22

TOE Identification – IBM WebSphere Message Broker, Version 6.0.0.3

EAL – Evaluation Assurance Level (EAL) 4 augmented with ALC_FLR.2

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, August 2005.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - EAL 4 augmented with ALC_FLR.2

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.

- Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
- Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Terminology

This section provides an explanation of each unique term used throughout this ST.

Message A string of bytes that is meaningful to the applications that use it. Messages are used to transfer information from one application program to another (or between different parts of the same application).

Queue Manager Responsible for maintaining the queues it owns and for storing all the messages it receives onto the appropriate queues.

1.3.3 Acronyms

This section provides a definition of each acronym used throughout this ST.

ACL	Access Control List
CC	Common Criteria
CM	Configuration Management
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
ID	Identification/Identifier
IT	Information Technology
MQ	Message Queue
OS	Operating System
SAR	Security Assurance Requirements
SF	Security Function
SFP	SF Policy
SFR	Security Functional Requirements
SOF	Strength of Function
SSL	Secure Socket Layer
ST	Security Target
TCP	Transmission Control Protocol

TLS	**Transport Layer Security**
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
TSS	TOE Summary Specification
WMB	WebSphere Message Broker

2 TOE Description

The Target of Evaluation (TOE) is IBM WebSphere Message Broker, Version 6.0.0.3.

WebSphere Message Broker (WMB) enables information, packaged as messages, to flow between different business applications, ranging from large legacy systems to unmanned devices such as sensors on pipelines. WMB provides the routing and data transformations necessary for the applications to communicate with one another. Communications between an application and WMB are via the WebSphere Message Queue (MQ) transport[1].

There are two ways in which WMB can act on messages:

1) **Message routing** from sender to recipient based on the content of the message where WMB can be configured for message routing via message flows that can be designed. A message flow describes the operations to be performed on the incoming message, and the sequence in which they are carried out. Each message flow consists of: A series of steps used to process a message, as defined in message flow nodes, and connections between the nodes that define the routes through the processing. Connections are made using message flow node connections. WMB provides built-in nodes and samples for numerous common functions. Additional functions can be built using a simple Graphical User Interface (GUI) to create user-defined nodes.
2) **Message transformation** before being delivered from one format to another by modifying, combining, adding or removing data fields. Transformations can be made by various nodes in a message flow but before a message flow node can operate on an incoming message, it must understand the structure of that message such as: some messages contain a definition of their own structure and format known as *self-defining messages* that can be handled without the need for additional information about the structure and format; and, other messages do not contain information about their structure and format. To process the later, a *message definition* of their structure must be created and made available. The message definitions defined are created within a *message set* which contains one or more message definitions. Like message flows, message definitions are built using GUI actions that contain two types of information: the *logical structure*—the abstract arrangement and characteristics of the data, represented as a tree structure; and, one or more *physical formats*—the way the data is represented and delimited in the physical bitstream.

2.1 TOE Overview

The TOE is comprised of the WMB components created by IBM. The TOE architecture consists of three subsystem functional components, which are placed at key points within the Enterprise architecture: Message Brokers Toolkit, Broker, and Configuration Manager. Additionally, the IT environment consists of the Applications (Clients) which use the TOE, databases used by the TOE, a repository used by the TOE, the MQ transport, the Java Runtime Environment (JRE), and the underlying operating systems. Figure 1 below shows these components (excluding the OS), their location, and interaction in the architecture.

[1] When the MQ transport is used, the IT environment provides message encryption.

WMB employs a distributed architecture. The following communication flows use the MQ transport mechanism which protects the flows from disclosure and modification: the Applications and Broker flow, the Broker and Configuration Manager flow, and the Configuration Manager and Message Brokers Toolkit flow.

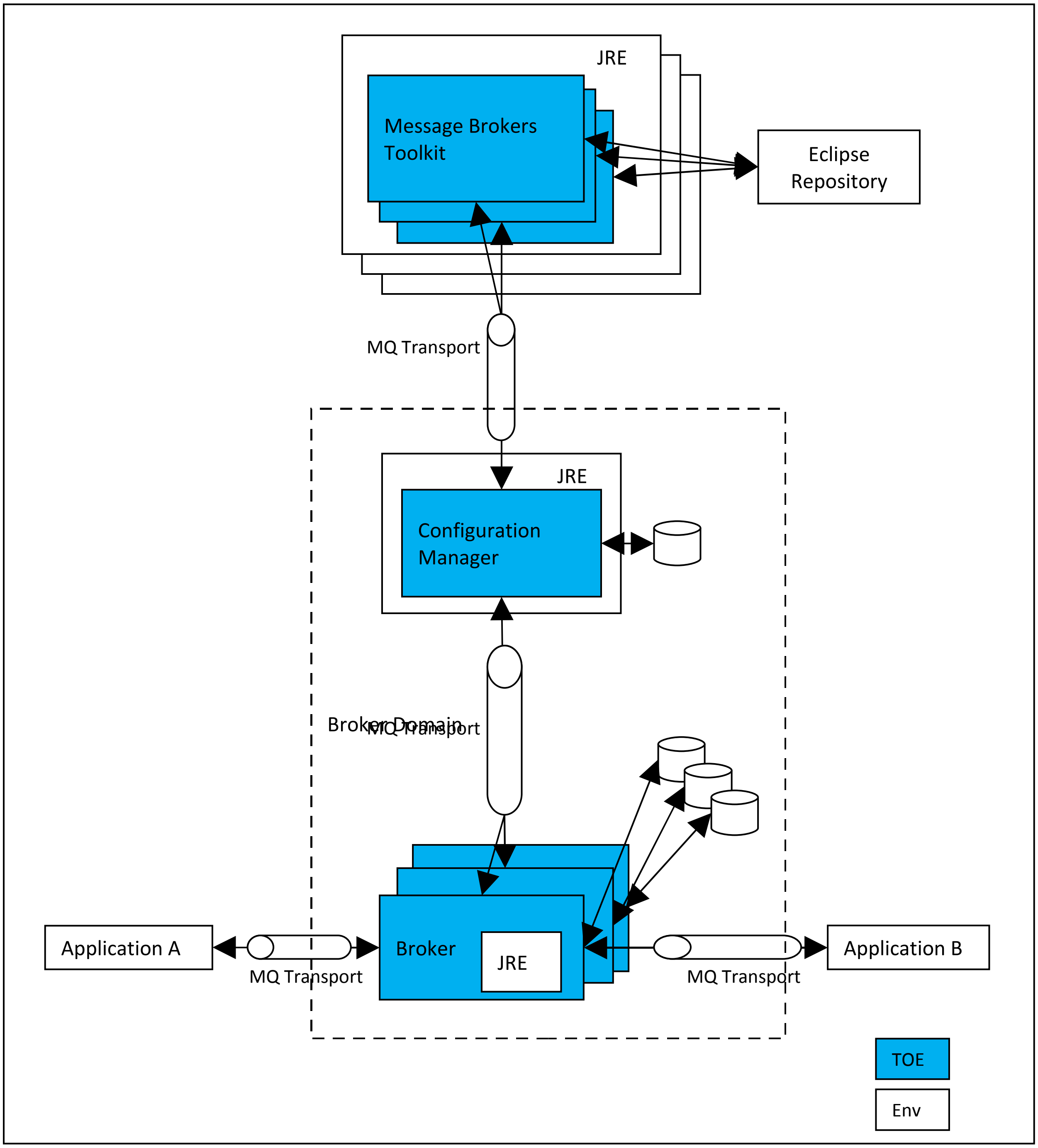

Figure 1 – WMB Components

2.2 TOE Architecture

Brokers are logically divided into Broker Domains. A collection of Brokers, that share a common Configuration Manager, form a broker domain. The Broker and Configuration Manager components run within a specific Broker

Domain environment and work together to enforce the overall TOE security policies. The Message Brokers Toolkit component utilizes the business integration repository.

2.2.1 Physical Boundaries

Each of the TOE components is a software application designed to execute within an operating system context provided by the environment. The Message Brokers Toolkit is an Eclipse application – which is based on Rational Application Developer version 6. The Configuration Manager is a Java application with a C++ wrapper. The Broker is a C/C++ application that uses JRE to perform some of its functions.

The evaluated configuration is supported on the following operating systems:

- AIX 5.3 – RS/6000 (POWER)
- HP-UX 11i – PA-RISC
- Red Hat Enterprise Linux (RHEL) AS4 for IA32
- RHEL AS4 for POWER
- RHEL AS4 for zSeries
- Solaris 9 - SPARC
- Windows XP Pro
- Windows Server 2003 Standard Edition, Enterprise Edition, and R2

The evaluated configuration requires a database that uses the ODBC protocol. The supported databases are:

- IBM DB2 v9.1 except on HP-UX
- IBM DB2 v8.2 on HP-UX only

The evaluated configuration supports WebSphere MQ 6.0.2.0. MQ contains and uses IBM's Global Security Kit (GSKit) library for SSL v3 and TLS v1.0.

2.2.1.1 The Message Brokers Toolkit

The Message Brokers Toolkit is an integrated development environment and graphical user interface used for management. The Message Brokers Toolkit also communicates with one or more Configuration Managers, and is used to manage broker domains. In the evaluated configuration, the Message Brokers Toolkit must be installed on the same machine as the Configuration Manager.

When the Message Brokers Toolkit is started, a single window is displayed called the workbench window and it displays one or more perspectives. A perspective is a collection of views and editors that help users complete a specific task, or work with specific types of resource. The Toolkit is aware of the user identity that initiates the workbench window display. However, the Toolkit relies upon the IT environment to establish the identification and authentication for this user.

There are a number of different perspectives available, the most commonly used of which are;

- The Broker Application Development perspective – this perspective allows developers to create new message flows by selecting Nodes from a pallet and wiring them together graphically. These message flows can be designed perform a large number of different task, for instance altering message content, routing messages according to content, sender etc. One or more message flows can then be put into a Broker Archive (BAR) file for deployment to one or more brokers.
- The Broker Administration Perspective – this perspective is used to connect to one or more Configuration Managers in order to administer the Broker domains which they control. From this perspective, the user can perform tasks such as creating + deleting execution groups, deploying message flows, starting and stopping message flows etc.

2.2.1.2 The Broker

A broker is a system service on Windows platforms or a daemon process on UNIX platforms that controls processes that run message flows.

Applications send messages to the broker. The broker routes each message from one application to another using the rules defined in message flows and message sets, and transforms the data into the structure required by the receiving application. Each message flow consists of one or more nodes connected together. Each node performs a processing step on the input data before passing the data to the next node in the message flow. A simple example of this would be an order processing application which has a single input node for all orders which could use a message flow to take messages from the input node and route the messages to different output nodes according to which warehouse needs to process the order. Each of these output nodes would then be serviced by a client application to deal with the order.

The TOE includes the following nodes:

- ESQLCompute – Extended Structured Query Language is a programming language to define and manipulate data within a message flow.
- Filter – Route a message according to message content.
- FlowOrder – Controls the order in which a message is processed by a message flow.
- Label – In combination with a RouteToLabel node, dynamically determines the route that a message takes through the message flow based on the message’s content.
- MQGet – Receives messages from clients that connect to the broker using MQ Transport.
- MQInput – Receives messages from clients that connect to the broker using MQ Transport.
- MQOutput – Sends messages to clients that connect to the broker using MQ Transport.
- MQReply – Sends a response to the originator of the input message.
- ResetContentDescriptor – Requests that a message is re-parsed by a different parser.
- RouteToLabel – In combination with a Label node, dynamically determines the route that a message takes through the message flow based on the message’s content.
- Throw – Throws an exception within a message flow.
- Trace – Generates trace records to help monitor the behavior of the message flow.
- TryCatch – Provides a special handler for exception processing.

The broker uses sender and receiver channels to communicate with the Configuration Manager and other brokers in the broker domain.

The broker is created using command line instructions on the machine where the component is installed. Brokers can be installed and created on one or more machines.

The broker depends on a broker database to hold broker information that includes control data for resources defined to the broker (i.e., deployed message flows). A database is defined and authorized access for specific users is created before creating the broker since creating the broker creates tables within the database (also known as the broker's local persistent store).

The broker connects to the database using an Open Database Connectivity (ODBC) connection.

When creating the Broker, a unique name within the broker domain must be identified. Broker names are case-sensitive on all supported platforms, except Windows platforms; and the same name must be used when creating a reference to the broker in the broker domain topology in the workbench. The reference to the broker is a representation of the physical broker in the configuration repository.

After creating the broker reference, changes are deployed to the broker domain. Deployment starts communications between the broker and the Configuration Manager. The broker receives configuration information from the

Configuration Manager, and stores it in the broker database. Deployment also initializes the broker to make it ready to execute message flows.

When a broker is created, a set of database tables for storing the information used by the broker to process messages at runtime is also defined and created.

2.2.1.3 The Configuration Manager

The Configuration Manager is the runtime component that acts as an intermediary between the Message Brokers Toolkit Broker Administration Perspective and the runtime broker domain. It is able to police which users of the Toolkit and Command-line tools are able to perform actions within the domain. In the evaluated configuration, the Configuration Manager will only be supported on Windows XP Pro.

In order to support both group level and user level security the following design is used by the Configuration Manager:

1) The object type, unique identifier, user name, machine/Windows domain name and requested action are flowed in a WebSphere MQ message from the Toolkit to the Configuration Manager. This flow can be SSL/TLS protected using the facilities provided by the underlying MQ transport.

2) The Configuration Manager retrieves the object type, name and requested action from the MQ message and passes the information onto one of the Configuration Manager's internal components (the *RoleManager* class) where the logic to grant or deny the requested operation is performed.

3) The group membership for the supplied user is discovered from the Operating System (OS).

4) The Access Control List (ACL) table in the Configuration Manager database is queried to see whether, based on the permission set configured by the administrator, permission is to be granted. Permissions can be granted to individual users or groups.

2.2.2 Logical Boundaries

The TOE logically supports the following security functions at its interfaces:

- Communication
- Security Audit
- Security Management
- Protection of the TSF
- User Data Protection

2.2.2.1 Communication

WMB provides message flow control of messages flowing through the broker. The nodes control the routing of messages (by the way they're connected within a message flow and by routing decisions made within certain nodes) and the transformation of messages.

2.2.2.2 Security Audit

WMB performs security auditing for all Toolkit Policy accesses made to the TOE. Audit records are generated when audit events occur. The audit records include the responsible user, date, time, and other details. The audit data is recorded into the operating system for protection.

2.2.2.3 Security Management

WMB provides security management functionality for the management of the access control policies. Management is performed from the command line.

2.2.2.4 Protection of the TSF

WMB protects itself and ensures that its policies are enforced in a number of ways. First, WMB interacts with users through well-defined interfaces designed to ensure that the WMB security policies are always enforced. Next, WMB encrypts all communications between physically separate parts of the TOE to ensure that no data is disclosed or modified.

2.2.2.5 User Data Protection

WMB protects user data by providing access control lists (ACLs) which mediate access between users and WMB objects.

2.3 TOE Documentation

IBM offers a series of documents that describe the installation process for WMB as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6 for information about these and other documents associated with WMB.

3 Security Environment

The security environment for the functions addressed by this specification includes threats and assumptions, as discussed below.

3.1 Threats

T.ADMIN_ERROR An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

T.ROLE A non-privileged user may gain administrative privileges and bypass the security policy gaining access to protected TOE resources.

T.TSF_COMPROMISE A malicious user or process may cause TSF data (including TOE audit data) and executable code to be inappropriately accessed (viewed, modified or deleted) by changing access to this data.

T.UNAUTH_ACCESS A user may gain unauthorized access (view, modify, delete) to user data by bypassing access controls or by intercepting data in a message flow.

T.UNDETECTED_ACTIONS A malicious user may cause the IT environment to fail to detect and record TOE generated audit events.

3.2 Assumptions

A.INSTALL_REQ The Message Brokers Toolkit and other applications that connect to the Configuration Manager through the Configuration Manager proxy must be installed on the same machine as the Configuration Manager.

A.NETWORK Data transferred between TOE components is protected from disclosure and modification by the IT Environment.

A.NO_EVIL Authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance. Also, message flow developers are trustworthy.

A.PHYSICAL It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

A.PLATFORM The IT Environment underlying the TOE is assumed to fulfill the requirements for the IT Environment described in this ST. It is also assumed that the IT Environment will provide a suitable operational environment for the TOE where the TOE will be able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled.

4 Security Objectives

This section defines the security objectives for the TOE and its environment. These objectives are suitable to counter all identified threats and cover all identified assumptions. The TOE security objectives are identified with 'O.' inserted at the beginning of the name and the environment objectives are identified with 'OE.' inserted at the beginning of the name.

4.1 Security Objectives for the TOE

O.ACCESS The TOE will ensure that users and applications gain only authorized access to it and to the resources that it controls.

O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.

O.AUDIT_GENERATION The TOE will provide the capability to create records of security relevant events associated with users.

O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators and authorized developers in their management of the security of the TOE.

O.TOE_PROTECTION The TOE will protect itself and its assets from external interference or tampering.

4.2 Security Objectives for the IT Environment

OE.AUDIT_STORAGE The IT environment will provide the capability to protect audit information.

OE.USER_AUTHENTICATION The IT environment will verify the claimed identity of users.

OE.USER_IDENTIFICATION The IT environment will uniquely identify users.

OE.SECURE_TRANS The IT environment will secure transmission between TOE components so that unauthorized user or process are unable to eavesdrop on communications between separate parts of the TOE allowing them to intercept and modify transmitted data.

OE.TIME The IT environment will provide a time source that provides reliable time stamps.

OE.TOE_PROTECTION The IT environment will provide protection to the TOE and its assets from external interference or tampering.

OE.PLATFORM The IT Environment underlying the TOE must fulfill the requirements for the IT Environment described in this ST. The IT Environment must provide a suitable operational environment for the TOE where the TOE is able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled.

4.3 Security Objectives for the Non-IT Environment

OE.ADMIN_GUIDANCE Administrators will be well trained and follow the administrative guidance supplied with the TOE.

OE.CONFIG The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation and applicable security policies and procedures. Message flows must be developed by trustworthy developers.

OE.INSTALL The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.

OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

OE.SELF_PROTECTION IT environment and its assets will be protected from external interference, tampering or unauthorized disclosure.

5 IT Security Requirements

The TOE makes no strength of function claim.

5.1 Extended Component Definitions

5.1.1 FAU_GEN_(EXP).1

This Security Target defines an extended component FAU_GEN_(EXP).1 as part of the FAU_GEN family of CC Part 2 for usage within this Security Target. This extended component is identical to FAU_GEN.1 except that it requires only failure outcomes to be audited instead of both success and failure outcomes as specified in FAU_GEN.1.2.

Component leveling

FAU_GEN_(EXP).1 Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

Management: FAU_GEN_(EXP).1

There are no management activities foreseen.

Audit: FAU_GEN_(EXP).1

There are no auditable events foreseen.

FAU_GEN_(EXP).1 Audit Data Generation on Failure

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN_(EXP).1.1 **The TSF shall be able to generate an audit record of the following auditable events:**

a) **Start-up and shutdown of the audit functions;**

b) **All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and**

c) **[assignment: *other specifically defined auditable events*].**

FAU_GEN_(EXP).1.2 **The TSF shall record within each audit record at least the following information:**

a) **Date and time of the event, type of event, subject identity, and the outcome (failure) of the event; and**

b) **For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*].**

5.2 TOE Security Functional Requirements

The following table describes the Security Functional Requirements (SFRs) that are candidates to be satisfied by WebSphere Message Broker.

Table 1 – TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN_(EXP).1: Audit Data Generation on Failure
	FAU_GEN.2: User Identity Association
FDP: User Data Protection	FDP_ACC.2: Complete Access Control
	FDP_ACF.1: Security Attribute Based Access Control
	FDP_IFC.2: Complete Information Flow Control
	FDP_IFF.1: Simple Security Attributes
FMT: Security Management	FMT_MSA.1a: Management of Security Attributes
	FMT_MSA.1b: Management of Security Attributes
	FMT_MSA.3a: Static Attribute Initialization
	FMT_MSA.3b: Static Attribute Initialization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security Roles
FPT: Protection of the TSF	FPT_RVM.1a: Non-bypassability of the TSP

5.2.1 Security Audit (FAU)

5.2.1.1 Audit Data Generation on Failure (FAU_GEN_(EXP).1)

FAU_GEN_(EXP).1.1 The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the [***not specified***] level of audit; and c) [**the following auditable events:**

-
- **Access Denied**
-
-
-
-
-
- **Repository Not Available**].
-
-

FAU_GEN_(EXP).1.2 The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity, and the outcome (failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**no additional details**].

Application Note: WMB writes event records to the operating system's logging facilities. The logging facilities supply the data and time for each record.

5.2.1.2 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.2 User Data Protection (FDP)

5.2.2.1 Complete Access Control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the [**Toolkit Access SFP**] on [**subjects: users; objects: configuration manager proxy, topology proxy, broker proxy, execution group proxy, subscription proxy, root topic proxy]** and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.2.2.2 Security Attribute Based Access Control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**Toolkit Access SFP**] to objects based on the following: [**subject security attributes: operating system user or group identity of the process requesting access; object security attributes: access control list (ACL)**].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**if an ACL exists which grants the user or group the requested access, then the requested access is allowed and granted to all objects beneath it in the hierarchy, unless it is overridden by another ACL].**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [

- **if a user or group has been granted access to a child object, they will receive implicit View access to the parent objects**
- **the user who creates/starts a Configuration Manager has full access control over the Configuration Manager]**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**no explicit denial**].

5.2.2.3 Complete Information Flow Control (FDP_IFC.2)

FDP_IFC.2.1 The TSF shall enforce the [**Message Flow SFP**] on [**originators and recipients of messages as subjects and the message flows**] and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

5.2.2.4 Simple Security Attributes (FDP_IFF.1)

FDP_IFF.1.1 The TSF shall enforce the [**Message Flow SFP**] based on the following types of subject and information security attributes: [**originators sending messages to a message flow and recipients receiving messages from a message flow where the security attributes in the message flow are defined by:**

- **the connections between each broker node (the output terminal and the input terminal of two broker nodes comprise a connection)**
- **the routing properties of the broker nodes**
- **the transformation properties of the broker nodes**].

Application Note: Each broker node has its own set of properties (connection properties, routing properties, and message transformation properties) which are documented in the guidance.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [

- **the message flow routes the message (or a modified version of the message) to the recipient based on the message flow rule set defined by the security attributes.**
- **the message flow transforms the message based on the message flow rule set defined by the security attributes**].

Application Note: Nodes have the ability to permit or deny the message flow based on the node's routing and connection properties.

FDP_IFF.1.3 The TSF shall enforce the [**none**].

FDP_IFF.1.4 The TSF shall provide the following [**none**].

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: [**none**].

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: [**none**].

5.2.3 Security Management (FMT)

5.2.3.1 Management of security attributes (FMT_MSA.1a)

FMT_MSA.1.1 The TSF shall enforce the **[Toolkit Access SFP]** to restrict the ability to **[manage]** the security attributes **[of Toolkit objects, namely ACLs]** to **[authorized administrators]**.

5.2.3.2 Management of security attributes (FMT_MSA.1b)

FMT_MSA.1.1 The TSF shall enforce the **[Message Flow SFP]** to restrict the ability to **[view, modify, or delete]** the security attributes **[of a message flow]** to **[authorized administrators and authorized developers]**.

5.2.3.3 Static Attribute Initialization (FMT_MSA.3a)

FMT_MSA.3.1 The TSF shall enforce the **[Toolkit Access SFP]** to provide **[*restrictive*]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[authorized administrator and authorized developer]** to specify alternative initial values to override the default values when an object or information is created.

5.2.3.4 Static Attribute Initialization (FMT_MSA.3b)

FMT_MSA.3.1 The TSF shall enforce the **[Message Flow SFP]** to provide **[*restrictive*]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[authorized administrator and authorized developer]** to specify alternative initial values to override the default values when an object or information is created.

5.2.3.5 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [

- **management of Toolkit Access SFP object attributes**
- **management of message flows]**.

5.2.3.6 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [

- **authorized administrator**
- **authorized developer**].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.2.4 *Protection of the TSF (FPT)*

5.2.4.1 Non-bypassability of the TSP (FPT_RVM.1a)

FPT_RVM.1.1a The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.3 IT Environment Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the IT environment of WebSphere Message Broker.

Table 2 – IT Environment Security Functional Components

Requirement Class	Requirement Component
FAU: Security Audit	FAU_STG.1: Protected Audit Trail Storage
FIA: Identification and Authentication	FIA_ATD.1 User Attribute Definition
	FIA_UAU.2: User Authentication Before any Action
	FIA_UID.2: User Identification Before any Action
FPT: Protection of the TSF	FPT_ITT.1: Basic Internal TSF Data Transfer Protection
	FPT_RVM.1b: Non-bypassability of the TSP
	FPT_SEP.1: TSF Domain Separation
	FPT_STM.1: Reliable Time Stamps

5.3.1 *Security Audit (FAU)*

5.3.1.1 Protected Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1 The ~~TSF~~ **IT environment** shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The ~~TSF~~ **IT environment** shall be able to [***prevent***] unauthorised modifications to the audit records in the audit trail.

5.3.2 *Identification and Authentication (FIA)*

5.3.2.1 User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1 The ~~TSF~~ **IT environment** shall maintain the following list of security attributes belonging to individual users**: operating system user identifier, operating system group identifier**.

5.3.2.2 User Authentication Before any Action (FIA_UAU.2)

FIA_UAU.2.1 The ~~TSF~~ **IT environment** shall require each user to be successfully authenticated before allowing any ~~other~~ TSF-mediated actions on behalf of that user.

5.3.2.3 User Identification Before any Action (FIA_UID.2)

FIA_UID.2.1 The ~~TSF~~ **IT environment** shall require each user to identify itself before allowing any ~~other~~ TSF-mediated actions on behalf of that user.

5.3.3 *Protection of the TSF (FPT)*

5.3.3.1 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1 The ~~TSF~~ **IT environment** shall protect TSF data from [***disclosure* and *modification***] when it is transmitted between separate parts of the TOE.

5.3.3.2 Non-bypassability of the TSP (FPT_RVM.1b)

FPT_RVM.1.1b The ~~TSF~~ **IT environment** shall ensure that ~~TSP~~ **IT environment's security policy** enforcement functions are invoked and succeed before each function within the ~~TSC~~ **IT environment's scope of control** is allowed to proceed.

5.3.3.3 IT Environment Security Function Domain Separation (FPT_SEP.1)

FPT_SEP.1.1 The ~~TSF~~ **IT environment** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ~~TSF~~ **IT environment** shall enforce separation between the security domains of subjects in the ~~TSC~~ **IT environment scope of control**.

5.3.3.4 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The ~~TSF~~ **IT environment** shall be able to provide reliable time stamps for ~~its own~~ use **by the TSF**.

5.4 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.2 as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Table 3 – EAL 4 Augmented with ALC_FLR.2 Assurance Components

ASSURANCE CLASS	ASSURANCE COMPONENTS
ACM: Configuration Management	ACM_AUT.1: Partial CM automation
	ACM_CAP.4: Generation support and acceptance procedures
	ACM_SCP.2: Problem tracking CM coverage
ADO: Delivery and Operation	ADO_DEL.2: Detection of modification
	ADO_IGS.1: Installation, Generation, and Start-up Procedures
ADV: Development	ADV_FSP.2: Fully defined external interfaces
	ADV_HLD.2: Security Enforcing High-level Design
	ADV_IMP.1 Subset of the implementation of the TSF
	ADV_LLD.1: Descriptive low level design
	ADV_RCR.1: Informal Correspondence Demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance Documents	AGD_ADM.1: Administrator Guidance
	AGD_USR.1: User Guidance
ALC: Life Cycle Support	ALC_DVS.1: Identification of Security Measures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.2: Analysis of Coverage
	ATE_DPT.1: Testing: High-level Design
	ATE_FUN.1: Functional Testing
	ATE_IND.2: Independent Testing - Sample

Assurance Class	Assurance Components
AVA: Vulnerability Assessment	AVA_MSU.2: Validation of analysis
	AVA_SOF.1: Strength of TOE Security Function Evaluation
	AVA_VLA.2: Independent Vulnerability Analysis

6 TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security Audit

WMB provides its own audit mechanism. It writes audit records to the underlying operating system - in UNIX, these records are stored in the SYSLOG and in Windows they are stored in the Event Log.

The auditable actions are Toolkit Policy accesses. Each audit record identifies the event type, responsible user (i.e. userid), data and time of the event, an indication of success or failure, and other information specific to each audit event. Starting and stopping of audit function is implicitly audited as a result of stopping and starting product.

The Security audit function is designed to satisfy the following security functional requirement:

- FAU_GEN_(EXP).1: The audit events as well as the audit record content enumerated above represent the set of required events and information.
- FAU_GEN.2: Each audit record contains the responsible user identifier.

6.1.2 Communication

WMB implements a Message Flow Policy. The Message Flow Policy controls the flow of messages (information) between the originator and the recipient. Message flows must be developed by trustworthy developers to ensure that messages flow from the intended originator to the intended recipient. Authorized administrators and authorized developers then deploy the message flows.

Each message flow in the broker consists of a set of interconnected nodes. Some node types can route a message to one of several nodes based on the information contained within the message. Nodes can also perform transformations on a message and pass the transformed message to the next node. The security attributes for a node varies between the different node types. The guidance documentation documents the properties (security attributes) of the different types of nodes. The types of nodes supported by the evaluated configuration are listed in section 2.2.1.2.

The Communication security functions are designed to satisfy the following security functional requirements:

- FDP_IFC.2: All message flows between an originator and recipient plus all operations within the message flow are subject to the Message Flow Policy.
- FDP_IFF.1: The broker has message flows that define the connection between nodes, the routing of messages within the message flow, and the transformation of messages within the message flow.

6.1.3 User data protection

6.1.3.1 Toolkit Access Control Policy

WMB implements a Toolkit access control Policy. The Toolkit Policy controls access between users and the following objects:

- Configuration Manager Proxy – This is the root of the object hierarchy.
- Root Topic Proxy – A character string that describes the nature of the data that is published in a publish/subscribe system. The root topic may contain sub-topics.
- Subscriptions Proxy – A record that contains the information that a subscriber passes to its local broker to describe the publications that it wants to receive.
- Topology Proxy - The brokers and collectives (and connections between them) in the broker domain.
- Broker Proxy - A set of execution processes that host one or more message flows.
- Execution Group Proxy - A named process or set of processes within a broker in which message flows are executed. The broker is guaranteed to enforce some degree of isolation between message flows in distinct execution groups because it ensures that they execute in separate address spaces, or as unique processes.

Access to all Toolkit objects is controlled by an access control list (ACL). An ACL entry contains a principal (a user identifier or a group identifier) and a permission. Access is determined by the corresponding permission on the first entry in the ACL that the user ID or group ID of the requesting user matches.

The User Data Protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.2: All users are subject to the Toolkit access policy for all available operations on configuration manager proxy, topology, broker, and execution group objects.

- FDP_ACF.1: Toolkit objects have ACLs. ACLs are compared against user identities and group memberships for that user in order to determine whether the request operation should be allowed.

6.1.4 Security Management

The TSF provides the ability to manage the security functions of the TOE. The authorized administrator is the only user permitted to perform management functions on a given object. An authorized administrator is anyone who has a Full-Control access control entry assigned to the object in question.

The management functions include the ability to manage the Toolkit access policies, including the ACLs associated with the controlled objects. Management occurs via a command line interface. By default, all Toolkit policy objects are created to deny access to everyone. When the Configuration Manager is created, the userid of the creator is given complete control over every object in the domain, which includes the ability to grant and deny access to others.

Message flows are created by authorized developers using restrictive default values and deployed by authorized administrators and authorized developers.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MSA.1a: The ability to manage controlled object attributes in the Toolkit Policy is restricted to an authorized administrator.
- FMT_MSA.1b: The ability to develop and deploy message flows as per the Message Flow Policy is restricted to authorized administrators and authorized developers.
- FMT_MSA.3a: By default every object covered by the Toolkit Policy is created with restricted access. The authorized administrator can change the access once the object has been created.
- FMT_MSA.3b: By default, every message flow created by an authorized developer and deployed by an authorized administrator/developer under the Message Flow Policy uses restrictive values.
- FMT_SMF.1: Administrators are able to perform all management functions, including management of Toolkit object attributes and management of message flows.

- FMT_SMR.1: Any user can be made an 'authorized administrator' by creating an Access Control Entry for that user for the Configuration Manager Proxy object with Full control permission. The creator of the Configuration Manager and the user identifier under which the Configuration Manager run automatically have an appropriate such entry created as required. Authorized developers create message flows for use by the broker.

6.1.5 TSF Self Protection

All communications between parts of the TOE are performed via the MQ transport. The MQ transport is outside the scope of this evaluation (but encrypted nonetheless). All encryption is performed using FIPS-validated algorithms. Furthermore, WMB has been carefully designed to offer well-defined interfaces that ensure that access to protected resources is subject to the applicable WMB security policies.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_RVM.1a: This requirement is met because the TOE security policy functions are invoked and succeed before each function within the TSC is allowed to proceed

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by IBM ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. IBM ensures changes to the implementation representation are controlled. IBM performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, and configuration management documentation and all of these items are identified in the CM Plan as configuration items.

These activities are documented in:

- WebSphere Message Broker - Configuration Management Plan

The Configuration management assurance measure satisfies the following EAL 4:

- ACM_AUT.1
- ACM_CAP.4
- ACM_SCP.2

6.2.2 Delivery and operation

IBM provides delivery documentation and procedures to identify the TOE, secure the TOE during delivery, and provide necessary installation and generation instructions. IBM's delivery procedures describe all applicable procedures to be used to prevent in appropriate access to the TOE. IBM also provides documentation that describes the steps necessary to install WebSphere Message Broker in accordance with the evaluated configuration.

These activities are documented in:

- WebSphere Message Broker - Installation and Delivery Guide

The Delivery and operation assurance measure satisfies the following EAL 4:

- ADO_DEL.2
- ADO_IGS.1

6.2.3 Development

IBM has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; a low-level design that describes the implementation of the subsystems; an implementation that contains the source code files; an informal security policy model; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the subsystems.

These activities are documented in:

- WebSphere Message Broker - Functional Specification
- WebSphere Message Broker - High-level Design
- WebSphere Message Broker - Design Correspondence Analysis
- WebSphere Message Broker – Low-level Design
- WebSphere Message Broker – Security Policy Model
- The implementation is documented in the source code files

The Development assurance measure satisfies the following EAL 4:

- ADV_FSP.2
- ADV_HLD.2
- ADV_IMP.1
- ADV_LLD.1
- ADV_RCR.1
- ADV_SPM.1

6.2.4 Guidance documents

IBM provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- WebSphere Message Broker - Administration Guide
- WebSphere Message Broker - User Guide

The Guidance documents assurance measure satisfies the following EAL 4:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

IBM ensures the adequacy of the procedures used during the development and maintenance of the TOE through its life-cycle. IBM includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure operation of the TOE. In addition, IBM identifies and tracks reported flaws, ensuring that they are addressed and corrections and corrective measures are made available as applicable.

These activities are documented in:

- WebSphere Message Broker - Life-cycle Plan

The Life cycle support assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ALC_DVS.1
- ALC_LCD.1
- ALC_TAT.1
- ALC_FLR.2

6.2.6 Tests

IBM has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. IBM has documented each test as well as an analysis of test coverage demonstrating that the security aspects of the design evident from the functional specification and high level design are appropriately tested. Actual test results are provided that demonstrate that the tests have been applied and that the TOE operates as designed. The test documentation consists of the following documents:

- WebSphere Message Broker - Test Plan
- WebSphere Message Broker - Test Coverage Analysis
- WebSphere Message Broker - Test Results

The Tests assurance measure satisfies the following EAL 4:

- ATE_COV.2
- ATE_DPT.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of WebSphere Message Broker and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references.

IBM performs regular vulnerability analyses of the entire TOE (including documentation) to identify obvious weaknesses that can be exploited in the TOE.

These activities are documented in:

- WebSphere Message Broker - Vulnerability Analysis Report

The Vulnerability assessment assurance measure satisfies the following EAL 4:

- AVA_MSU.2
- AVA_SOF.1
- AVA_VLA.2

7 Protection Profile Claims

There are no Protection Profile claims.

8 Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Security Functional Requirement Dependencies;
- Explicitly Stated Requirements;
- TOE Summary Specification; and
- PP Claims.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of threats and usage assumptions by the security objectives.

	T.ADMIN_ERROR	**T.ROLE**	**T.TSF_COMPROMISE**	**T.UNAUTH_ACCESS**	**T.UNDETECTED_ACTIONS**	**A.INSTALL_REQ**	**A.NETWORK**	**A.NO_EVIL**	**A.PHYSICAL**	**A.PLATFORM**
O.ACCESS				X						
O.ADMIN_ROLE		X								
O.AUDIT_GENERATION					X					
O.MANAGE	X									
O.TOE_PROTECTION			X							
OE.AUDIT_STORAGE			X							
OE.SECURE_TRANS							X			
OE.TIME					X					
OE.TOE_PROTECTION			X				X			
OE.ADMIN_GUIDANCE	X									
OE.CONFIG	X					X		X		
OE.INSTALL	X					X				

	T.ADMIN_ERROR	T.ROLE	T.TSF_COMPROMISE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	A.INSTALL_REQ	A.NETWORK	A.NO_EVIL	A.PHYSICAL	A.PLATFORM
OE.PHYSICAL			X		X				X	
OE.SELF_PROTECTION				X						
OE.PLATFORM					X					X
OE.USER_AUTHENTICATION		X								
OE.USER_IDENTIFICATION		X								

Table 4 Environment to Objective Correspondence

8.1.1.1 T.ADMIN_ERROR

An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is satisfied by ensuring that:

- O.MANAGE: Improper administration could result if the TOE does not provide the proper administration tools. There is always the possibility that the administrator will make an honest mistake. This threat should be mitigated as long as the TOE provides the necessary administrator support.
- OE.ADMIN_GUIDANCE: Improper administration could result if the authorized administrator is unknowledgeable. There is always the possibility that the administrator will make an honest mistake. This threat should be mitigated as long as the authorized administrator is provided with knowledge necessary to carry out administrative duties.
- OE.INSTALL: The authorized administrator is provided with necessary installation instructions from the product developer that details how to securely install the TOE.
- OE.CONFIG: The authorized administrator is provided with necessary instructions to securely configure the TOE.

8.1.1.2 T.ROLE

A non-privileged user may gain administrative privileges and bypass the security policy gaining access to protected TOE resources.

This Threat is satisfied by ensuring that:

- O.ADMIN_ROLE: This requires the TOE to support the concept of an Administrator to manage the TOE.
- OE.USER_AUTHENTICATION: This requires users to authenticate their identity prior to accessing the TOE.
- OE.USER_IDENTIFICATION: This requires users to claim their unique identity prior to accessing the TOE.

8.1.1.3 T.TSF_COMPROMISE

A malicious user or process may cause TSF data (including TOE audit data) and executable code to be inappropriately accessed (viewed, modified or deleted) by changing access to this data.

This Threat is satisfied by ensuring that:

- OE.AUDIT_STORAGE: The IT environment will protect the audit data.
- O.TOE_PROTECTION: The TSF data and executable code is protected under the TOE objective for TOE protection.
- OE.TOE_PROTECTION: The TSF data and executable code is protected under the environmental objective for TOE protection.
- OE.PHYSICAL: The IT environment will protect the TSF data and executable code from a compromise through physical means.

8.1.1.4 T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data by bypassing access controls or by intercepting data in a message flow.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE must satisfy the objective of ensuring that only authorized users may gain access to the TOE and the resources it protects, and that users are not allowed to access protected data for which they are not authorized.
- OE.SELF_PROTECTION: The IT environment and its assets are protected under the environmental objective for self-protection.

8.1.1.5 T.UNDETECTED_ACTIONS

A malicious user may cause the IT environment to fail to detect and record TOE generated audit events.

This Threat is satisfied by ensuring that:

- O.AUDIT_GENERATION: Non-physical security relevant actions are detected and a record created and provided to the IT environment.
- OE.TIME: All audit records include reliable timestamps.
- OE.PLATFORM: The IT Environment must provide a suitable operational environment for the TOE to properly execute and the IT Environment must properly fulfill the dependencies the TOE has upon the IT Environment.
- OE.PHYSICAL: The threat of undetected physical manipulation of the TOE is addressed by the physical protection in the environment.

8.1.1.6 A.INSTALL_REQ

The Message Brokers Toolkit and other applications that connect to the Configuration Manager through the Configuration Manager proxy must be installed on the same machine as the Configuration Manager.

This Assumption is satisfied by ensuring that:

- OE.CONFIG: Authorized administrators are trusted to properly configure the IT environment so it enforces its security policies.
- OE.INSTALL: Authorized administrators have the proper documentation to install configure and manage the TOE and will follow that documentation.

8.1.1.7 A.NETWORK

Data transferred between TOE components is protected from disclosure and modification by the IT Environment.

This Assumption is satisfied by ensuring that:

- OE.SECURE_TRANS : The IT environment will secure transmission between the TOE components so that unauthorized user or process are unable eavesdrop on communications between separate parts of the TOE allowing them to intercept and modify transmitted data
- OE.TOE_PROTECTION: The IT environment will provide protection of the TSF data from disclosure and modification.

8.1.1.8 A.NO_EVIL

Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance. Also, message flow developers are trustworthy.

This Assumption is satisfied by ensuring that:

- OE.CONFIG: Authorized administrators are trusted to properly configure the TOE and IT environment so it enforces its security policies. Messages flows added to the TOE must come from trustworthy developers.

8.1.1.9 A.PHYSICAL

It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.

8.1.1.10 A.PLATFORM

The IT Environment underlying the TOE is assumed to fulfill the requirements for the IT Environment described in this Security Target. It is also assumed that the IT Environment will provide a suitable operational environment for the TOE where the TOE will be able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled.

This Assumption is satisfied by ensuring that:

- OE.PLATFORM: This objective basically reiterates the assumption to expect the IT Environment to provide a suitable and effective environment for the operation of the TOE.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that Table 5 indicates the requirements that effectively satisfy the individual objectives. .

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

	O.ACCESS	**O.ADMIN_ROLE**	**O.AUDIT_GENERATION**	**O.MANAGE**	**O.TOE_PROTECTION**	**OE.SECURE_TRANS**	**OE.AUDIT_STORAGE**	**OE.TIME**	**OE.TOE_PROTECTION**	**OE..USER_AUTHENTICATION**	**OE.USER_IDENTIFICATION**
FAU_GEN_(EXP).1			X								

	O.ACCESS	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.MANAGE	O.TOE_PROTECTION	OE.SECURE_TRANS	OE.AUDIT_STORAGE	OE.TIME	OE.TOE_PROTECTION	OE..USER_AUTHENTICATION	OE.USER_IDENNTIFICATION
FAU_GEN.2			X								
FDP_ACC.2	X										
FDP_ACF.1	X										
FDP_IFC.2	X										
FDP_IFF.1	X										
FMT_MSA.1a				X							
FMT_MSA.1b				X							
FMT_MSA.3a				X							
FMT_MSA.3b				X							
FMT_SMF.1				X							
FMT_SMR.1		X									
FPT_RVM.1a					X						
FAU_STG.1							X				
FIA_ATD.1											X
FIA_UAU.2										X	
FIA_UID.2											X
FPT_ITT.1						X			X		
FPT_RVM.1b									X		
FPT_SEP.1									X		
FPT_STM.1								X			

Table 5 Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE will ensure that users and applications gain only authorized access to it and to the resources that it controls.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.2: The subjects and objects within the TOE are under the enforcement of an access control policy. All operations between the subjects and objects are controlled by the access policies.
- FDP_ACF.1: The subjects and objects under the access control policies have certain rules that apply to all accesses between them. All accesses are controlled by decisions based on user identities and ACLS on objects.
- FDP_IFC.2: All messages in a message flow are under the enforcement of a message flow policy. All operations on a message are controlled by the message flow policies.
- FDP_IFF.1: The messages under the message flow policies have rules that apply to all operations on them.

8.2.1.2 O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMR.1: The TOE will establish an authorized administrator role and an authorized developer role. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to managing the security policies. The authorized developer will be able to developer message flows for the broker.

8.2.1.3 O.AUDIT_GENERATION

The TOE will provide the capability to create records of security relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN_(EXP).1: This objective is satisfied in part by the requirement that the TOE generate audit records.
- FAU_GEN.2: Each audit record written must be descriptive of the event that caused a record to be generated, and must be associated with the unique identity of the user that caused the event.

8.2.1.4 O.MANAGE

The TOE will provide all the functions and facilities necessary to support the authorized administrators and authorized developers in their management of the security of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MSA.1a: Only authorized administrators may manipulate the security attributes of Toolkit policy objects, namely ACLs.
- FMT_MSA.1b: Only authorized administrators and authorized developers may manage the security attributes of message flows.
- FMT_MSA.3a: Only authorized administrators may manipulate the security attributes of Toolkit policy objects.
- FMT_MSA.3b: Only authorized administrators and authorized developers may manipulate the security attributes of message flows.
- FMT_SMF.1: The authorized administrator will be able to manage the access policies object attributes. The authorized administrator and authorized developer will be able to manage the message flows.

8.2.1.5 O.TOE_PROTECTION

The TOE will protect itself and its assets from external interference or tampering.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1a: The TOE is required to allow access to protected objects only after it makes informed access decisions.

8.2.1.6 OE.AUDIT_STORAGE

The IT environment will provide the capability to protect audit information.

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_STG.1: The IT environment is required to protect the audit records from deletion.

8.2.1.7 OE.USER_AUTHENTICATION

The IT environment will verify the claimed identity of users.

This IT Environment Security Objective is satisfied by ensuring that:

- FIA_UAU.2: Users must be authenticated before they can perform any TSF-mediated functions.

8.2.1.8 OE.USER_IDENTIFICATION

The IT environment will uniquely identify users.

This IT Environment Security Objective is satisfied by ensuring that:

- FIA_ATD.1: The IT environment maintains identification information for individual users in the form of operating system security attributes.
- FIA_UID.2: Users must be identified to the IT environment before they can perform any TSF-mediated functions.

8.2.1.9 OE.SECURE_TRANS

The IT environment will secure transmission between the TOE components so that unauthorized user or process are unable eavesdrop on communications between separate parts of the TOE allowing them to intercept and modify transmitted data.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_ITT.1: The IT environment will protect all TSF data while it is in transit among distributed portions of the TOE.

8.2.1.10 OE.TIME

The IT environment will provide a time source that provides reliable time stamps.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The IT environment is required to provide a reliable time source.

8.2.1.11 OE.TOE_PROTECTION

The IT environment will provide protection to the TOE and its assets from external interference or tampering.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_RVM.1b: The IT environment is required to access to protected objects only after it makes informed access decisions.
- FPT_ITT.1: The IT environment will protect all TSF data while it is in transit among distributed portions of the TOE.
- FPT_SEP.1: The IT environment is required to protect itself and separate the contexts of its users.

8.3 Security Assurance Requirements Rationale

WMB is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a low attack potential. As such, EAL 4 is appropriate to provide the assurance necessary to counter the potential for attack.

8.4 Strength of Function Rationale

The TOE makes no strength of function claim.

8.5 Requirement Dependency Rationale

The following table represents an analysis of the dependencies of the security functional requirements (SFRs) in this security target. The first column identifies all of the SFRs in this security target. The TOE SFRs are highlighted in bold, unlike the IT environment SFRs. The second column identifies the minimum dependencies defined in the Common Criteria v2.3 and associated interpretations[2]. The third column identifies the actual requirements in this

[2] No International Interpretations are applicable to any SFR or SAR in this Security Target.

security target that correspond to the identified dependencies. Notice that this table demonstrates that all of the identified dependencies are satisfied.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN_(EXP).1	FPT_STM.1	*FPT_STM.1*
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN_(EXP).1 and FIA_UID.2
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2 and FMT_MSA.3a
FDP_IFC.2	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.2 and FMT_MSA.3b
FMT_MSA.1a	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2
FMT_MSA.1b	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_IFC.2
FMT_MSA.3a	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1a and FMT_SMR.1
FMT_MSA.3b	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1b and FMT_SMR.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_RVM.1a	none	none
FAU_STG.1	FAU_GEN.1	FAU_GEN_(EXP).1
FIA_ATD.1	none	none
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	none	none
FPT_ITT.1	none	none
FPT_RVM.1b	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none

Table 6 Requirement Dependencies

Since FAU_GEN_(EXP).1 is nearly identical to FAU_GEN.1, FAU_GEN_(EXP).1 is used as a substitute for FAU_GEN.1 in the table above.

8.6 Explicitly Stated Requirements Rationale

This security target includes no explicitly stated requirements.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. Table 7 Security Functions vs. Requirements Mapping demonstrates the relationship between security requirements and security functions.

	Security audit	Communications	User data protection	Security management	TSF Self Protection
FAU_GEN_(EXP).1	X				
FAU_GEN.2	X				
FDP_ACC.2			X		
FDP_ACF.1			X		
FDP_IFC.2		X			
FDP_IFF.1		X			
FMT_MSA.1a				X	
FMT_MSA.1b				X	
FMT_MSA.3a				X	
FMT_MSA.3b				X	
FMT_SMF.1				X	
FMT_SMR.1				X	
FPT_RVM.1a					X

Table 7 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.