WebSphere MQ
EAL2
Security Target

IBM Global Services CLEF
IBM UK Ltd
Meudon House
Meudon Avenue
Farnborough
Hampshire GU14 7NB

Telephone No: (01252) 558472

Date: 04th May 2004
Issue: 2.8
Reference: LFF/MQ/EAL2/ST/28

This Page Intentionally Left Blank.

Table of Contents

Glossary and Terminology

ACL	Access Control List
Administrator	A user with membership to the MQM administrator group within the operating system
Authorised User	A user who may, in accordance with the TSP, perform an operation.
CC	Common Criteria
Channel	Channels are objects that provide a communication path from one queue manager to another.
CSD	Corrective Service Diskette. This is a fix pack, which contains a collection of fixes and is cumulative e.g. all fixes within CSD5 are contained within CSD6.
DAP	Data Abstraction and Persistence
EAL	Evaluation Assurance Level
FIFO	First In First Out
IT	Information Technology
ITSEC	IT Security Evaluation Criteria
MCA	Message Channel Agent. A program that transmits prepared messages from a transmission queue to a communication link, or from a communication link to a destination queue.
Message	A *message* is 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).
MQ	Message Queue
MQI	Message Queue Interface
OAM	Object Authority Manager
Object	Objects are queues, process definitions and namelists.
OS	Operating System
OSP	Organisational Security Policy
Queue	A *queue* is a data structure used to store messages. Each queue is owned by a *queue manager*.
Queue Manager	The Queue Manager is responsible for maintaining the queues it owns, and for storing all the messages it receives onto the appropriate queues.
SF	Security Function. A part or parts of the TOE that have been relied upon for enforcing a closely related subset of rules from the TSP.
SFR	Security Functional Requirement

SOF	Strength of Function
ST	Security Target
TOE	Target Of Evaluation. An IT product or system and its associated administrator and user guidance documentation that is the subject of the evaluation.
TSF	TOE Security Function. A set consisting of all hardware, software and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
TSP	TOE Security Policy. A set of rules that regulate how assets are managed, protected and distributed within the TOE.

1 Introduction

This document is the Security Target (ST) for the Common Criteria (CC) evaluation of WebSphere MQ version 5.3.0.2 with Corrective Service Diskette (CSD) 6.

This document is the WebSphere EAL2 Security Target, version 2.8 and dated 4th May 2004.

1.1 Overview

IBM WebSphere® MQ is message queuing middleware. It connects all business software together to form one enterprise by providing an open, scalable, industrial-strength messaging backbone.

MQ is divided into the operating system specific editions as follows:

- WebSphere MQ for AIX;
- WebSphere MQ for HP-UX;
- WebSphere MQ for Linux Intel;
- WebSphere MQ for Linux zSeries;
- WebSphere MQ for Sun Solaris; and
- WebSphere MQ for Windows.

Each of the operating system specific Editions encompass the following components:

- MQ server (which includes the queue manager);
- MQ Client and
- MQI.

In addition, to the above components, there are tools and utilities to enable third party development of applications. These applications are often referred to as 'MQ applications' however these are not within the scope of the evaluation.

The following Operating Systems (OS) are supported within this evaluation:

- AIX v5.1;
- AIX v5.2;
- HP-UX 11i;
- SUSE Linux Enterprise Server 8 (for Linux Intel and Linux zSeries);
- RedHat Enterprise Linux AS 2.1 (for Linux Intel);
- Sun Solaris 8;
- Sun Solaris 9;

- Microsoft Windows 2000 (this includes all combinations of Advanced Server, Server, Professional, Service Packs and hotfixes); and
- Microsoft Windows 2003 (this includes all combinations of Standard, Enterprise, Service Packs and hotfixes).

It is assumed that all hardware used within the operating environment is secured such that no potential vulnerabilities could be introduced that would circumvent the functionality described within this ST.

1.2 Description

WebSphere MQ (WMQ) allows application programs to use *message queuing* to participate in message-driven processing. Application programs can communicate across different platforms by using WMQ. For example, AIX and Sun Solaris applications can communicate through WebSphere MQ. The applications are shielded from the mechanics of the underlying communications.

Messages are used to transfer information from one application program to another (or between different parts of the same application). The applications can be running on the same platform, or on different platforms.

Each queue is owned by a *queue manager*. The queue manager is responsible for maintaining the queues it owns, and for storing all the messages it receives onto the appropriate queues. The messages might be put on the queue by application programs, or by a queue manager as part of its normal operation.

The TOE is indicated by the dotted line shown in Figure 2.1 of this ST.

Figure 1.1 shows an example of how different servers and clients can communicate with one another. The operating systems on each of these can be independent to one another and the applications can reside on the same machine as the queue manager without the requirement for a client.

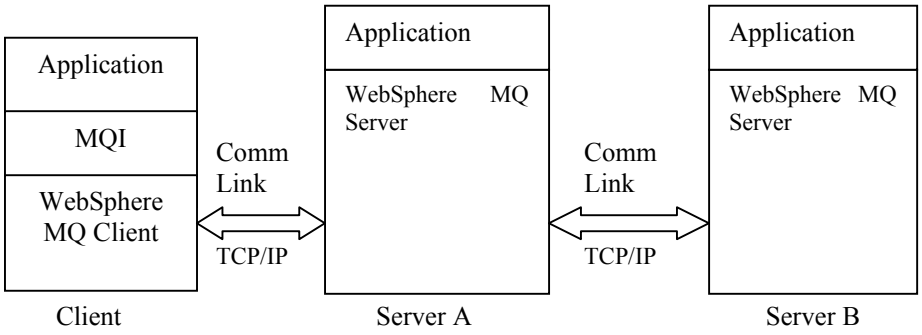

Figure 1.1 Connections between MQ servers and clients

1.2.1 WMQ Server

The WMQ server contains the queue manager, which is responsible for maintaining the queues that it owns, and for storing all the messages it receives onto the appropriate queues. The server also contains components that:

- Interface with the operating system to retrieve information (Common Services),

- Provides a command line interface for administration of the queues; and

- Interface to remote queue managers (Message Channel Agent (MCA)). This component is responsible for sending and receiving of messages to remote queues. Messages are transmitted between queue managers on a *channel*. *Channels* are objects that provide a communication path from one queue manager to another.

1.2.2 WMQ Client

MQ Client is part of the WebSphere product that can be installed on its own, on a separate machine from the server. An application can be run on a WMQ client and it can interact with one or more WMQ servers and can connect to their queue managers by means of a communications protocol. The servers to which the client connects might or might not be part of a cluster.

1.2.3 Message Queue Interface (MQI)

WMQ implements a component known as the *Message Queue Interface* (MQI) that provides a common application-programming interface wherever the applications run. This makes it easier to port application programs from one platform to another and enables MQI to be running on a separate machine to the queue manager. Applications use Programmable Command Format (PCF) commands to send calls to the Queue Manager via the MQI.

1.3 CC Conformance

This ST is [CC] *Part 2 extended with* FAU_GEN_MQ.1 and FMT_MSA_MQ.3 *and Part 3 conformant*, with a claimed Evaluation Assurance Level of EAL2.

1.4 Strength of Functions

There is no strength of function claim because the TOE does not identify any security functional requirements for which an explicit Strength Of Function (SOF) is appropriate and does not identify any functions that are of a permutational or probabilistic nature. Therefore, a minimum SOF claim is not included for the TOE.

1.5 References

[CC] Common Criteria for Information Technology Security Evaluation, CCIMB-99-031, Version 2.1, August 1999.

1.6 Structure

The structure of this document is as defined by [CC] Part 1, Annex C:

- Section 2 is the TOE description;

- Section 3 provides a statement of the TOE security environment;
- Section 4 provides the statement of IT security objectives;
- Section 5 provides a statement of IT security requirements;
- Section 6 provides the TOE summary specification, which includes the detailed specification of the IT functions; and
- Section 7 provides the rationale for the security objectives, security requirements and TOE summary specification.

2 TOE Description

The below diagram shows the physical scope and boundaries in relation to the components within the TOE. It should be noted that the TOE does not include the MQ Client or MQI components of the operating system specific editions. The TOE is a subset of the product and the dotted line within figure 2.1 illustrates the boundary of the TOE in relation to the components. It should be noted that the applications, application programs and operating systems referred to within this ST are outside the scope of the evaluation.

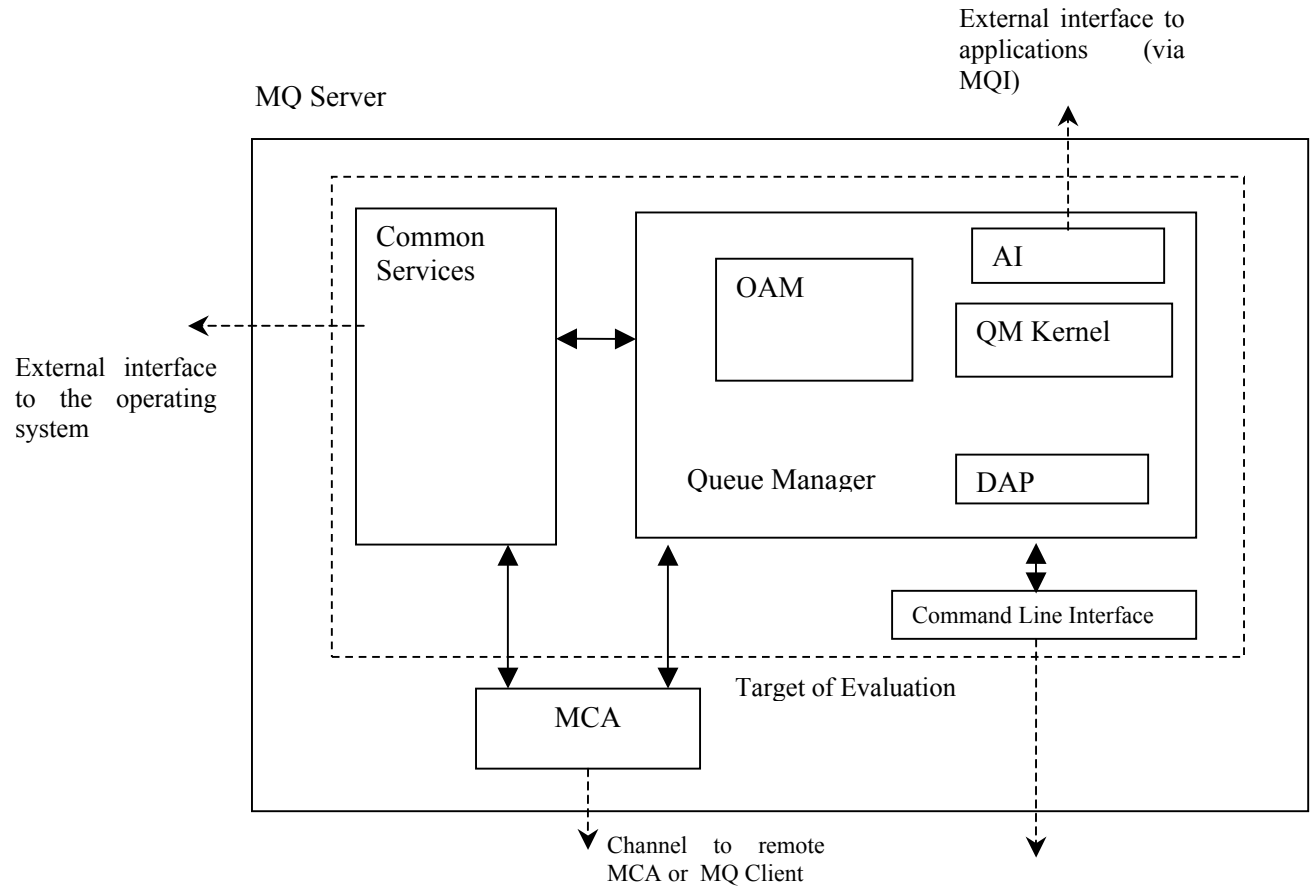

Figure 2.1: TOE boundary

2.1 Queue Manager

A *queue manager* is a system program that provides queuing services to applications. It provides an application-programming interface so that programs can put messages on, and get messages from, queues. A queue manager provides additional functions so that administrators can create new queues, alter the properties of existing queues, and control the operation of the queue manager. Some of the access control functions (e.g. creation

and deletion) on Queue Managers are performed by the OS and therefore management of the queue manager is not within the scope of the evaluation.

2.1.1 Object Authority Manager (OAM)

Authorization for using MQI calls, commands, and access to objects, is provided by the Object Authority Manager (OAM), which is enabled by default. Access to WebSphere MQ objects is controlled by the OAM based upon User and Group IDs controlled by the operating system. A command line interface to enable administrators to grant or revoke authorizations is used.

The OAM needs to be able to identify who is requesting access to a particular object. WebSphere MQ uses the term *principal* to refer to the User identifier associated with a user. The principal is established when the application first connects to the queue manager; it is determined by the queue manager from the user ID associated with the connecting application.

WebSphere MQ propagates the user ID received from the system in the message header of each message as identification of the user. This user ID is then checked against those on the Access Control List (ACL) of the object.

2.1.2 QM Kernel

In WebSphere MQ, event monitoring is performed by the QM Kernel. An instrumentation event is a logical combination of conditions that is detected by a queue manager. Such an event causes the queue manager or channel instance to put a special message, called an *event message*, on an event queue. An event queue is like any other MQ queue in the way access is controlled but it stores only event messages.

WebSphere MQ instrumentation events provide information about errors, warnings, and other significant occurrences in a queue manager, and in particular authorisation failures.

2.1.3 Application Interface (AI)

The AI component provides an external interface to the TOE. It is responsible for accepting calls from an application, and performing simple syntax checking on the parameters.

Applications can access MQ objects (queues, process definitions and namelists) by issuing MQI calls. The applications can also use PCF commands to access these objects.

2.1.4 Data Abstraction and Persistence (DAP)

The DAP component of the Queue Manager holds the attributes of objects such as process definitions and queues, and the messages on the queues. The DAP component is responsible for the local queue attributes. None of these are security attributes as defined within this ST.

2.2 Command line Interface

The command line interface is used to enable administrators to provide management of the queue manager. To access the command line interface, the user must be a member of

the *mqm* group on the OS. The *mqm* group needs to be created by the administrator prior to the installation of the TOE.

Administrators can use control commands to administer WMQ. One of these control commands is *setmqaut*, which is used to grant authorities to other users to enable them to access WMQ resources.

Administrators can use the control command *runmqsc* to enable the use of Message Queue Script Commands (MQSC). These are used to manage the message queues.

Another interface for Microsoft Windows is available called MQ Explorer, however this is not included within the scope of evaluation.

2.3 Common Services

The Common Services layer provides an Operating System (OS) independent, external interface to those services that other components wish to use that contains platform specific code.

2.4 TOE Environment

WMQ relies upon the OS to provide the security environment to protect both the client and server. The OS also provides WMQ with user and group IDs, time and date information so that WMQ can enforce its security functionality. In addition, in order that the audit records produced by the TOE can be read, an application within the environment is required.

Message Channel Agent is a program that transmits prepared messages from a transmission queue to a communication link, or from a communication link to a destination queue.

3 TOE Security Environment

3.1 Introduction

The statement of TOE security environment describes the security aspects of the environment in which it is intended that the TOE will be used and the manner in which it is expected to be employed.

The statement of TOE security environment therefore identifies the assumptions made on the operational environment and the intended method for the product, defines the threats that the product is designed to counter and the organisational security policies which the product is designed to comply.

3.2 Threats

The assumed security threats are listed below:

3.2.1 Threats countered by the TOE

[T.ACCESS_RES]	An authorised user of the TOE gains access to an object without the correct authority to access that object.
[T.ACCOUNT]	Unauthorised attempts to access objects for which the user has no authority go undetected.

3.2.2 Threats countered by the TOE Environment

[T.ACCESS_TOE]	An unidentified user gains access to the TOE and it's objects.
[T.ROLE]	A non-privileged user gains administrative privileges.
[T.NETWORK]	Data transferred between workstations is disclosed to, or modified by unidentified users or processes, either directly or indirectly.
[T.OS]	The operating system on which the TOE is installed becomes compromised.

3.3 Organisational Security Policies (OSPs)

The TOE complies with the following OSP:

[P.ACCESS] The right to access a specific object is determined on the basis of:

- The identity of the subject attempting to access the object; or
- Membership of a group that has access rights to the object.

3.4 Assumptions

This section provides the minimum physical and procedural measures required to maintain security of the WebSphere MQ product.

3.4.1 Physical aspects

[A.OS] It is assumed that the operating system has been configured in accordance with the manufacturer's installation guides and where applicable, in its evaluated configuration. It is securely configured such that the operating system protects the TOE from any unauthorised users or processes.

[A.PROTECT] It is assumed that all software and hardware, including network and peripheral devices, have been approved for the transmittal of protected data. Such items are to be physically protected against threats to the confidentiality and integrity of the data.

3.4.2 Personnel Aspects

[A.ADMIN] It is assumed that there are one or more competent individuals that are assigned to manage the TOE and the security of the information it contains. Such personnel are assumed not to be careless, wilfully negligent or hostile.

4 Security Objectives

4.1 Security Objectives for the TOE

[O.ACCESS] The TOE must ensure that only those users with the correct authority are able to access an object.

[O.ACCOUNT] The TOE must provide a means of recording any unsuccessful access attempts to the objects.

[O.MANAGE] The TOE must allow administrators to effectively manage the TOE and that this is only performed by authorised users.

4.2 Security Objectives for the TOE Environment

[O.IDENTIFY] The Operating System must ensure that all users are identified.

[O.ROLE] The operating system must be able to associate users with roles and maintain an *administrator* role.

[O.TIME] The operating system must ensure that the clock is accurate and reliable.

[O.ADMIN] Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

[O.CONFIG] Those responsible for the TOE must ensure that each user on the supporting operating system has a unique user ID and that the operating system is configured to ensure that only approved groups of users may access the system.

[O.OS] Those responsible for the TOE must ensure that the supporting operating system is installed and configured in accordance with the manufacturer's instructions, the evaluated configuration where applicable and is secure.

[O.PROTECT] Those responsible for the TOE must ensure that procedures and/or mechanisms exist to ensure that data transferred between workstations is secured from disclosure, interruption or tampering.

[O.RECOVER] Those responsible for the TOE must ensure that procedures and/or mechanisms are provided to ensure that after system failure or other discontinuity, recovery without a security compromise is obtained.

5 Security Requirements

This section specifies the Security Functional Requirements (SFRs) for the TOE and organises the SFRs by class.

Within the text of each SFR, the selection and assignment operations (as defined within [CC]) are *italicised*.

Note: FAU_GEN_MQ.1 and FMT_MSA_MQ.3 are explicitly stated IT security requirements, and have not been specified using CC Part 2 functional components.

The International Interpretations that have been applied for the Security Requirements are 058, 064, 065, and 103.

5.1 TOE Security Functional Requirements

The following table summarises the SFRs:

CLASS	FAMILY	COMPONENT	ELEMENT
FAU	FAU_GEN	FAU_GEN_MQ.1	FAU_GEN_MQ.1.1
			FAU_GEN_MQ.1.2
		FAU_GEN.2	FAU_GEN.2.1
	FAU_STG	FAU_STG.1	FAU_STG.1.1
			FAU_STG.1.2
FDP	FDP_ACC	FDP_ACC.1	FDP_ACC.1.1
	FDP_ACF	FDP_ACF.1	FDP_ACF.1.1
			FDP_ACF.1.2
			FDP_ACF.1.3
			FDP_ACF.1.4
FMT	FMT_MSA	FMT_MSA.1	FMT_MSA.1.1
		FMT_MSA_MQ.3	FMT_MSA_MQ.3.1
	FMT_MTD	FMT_MTD.1	FMT_MTD.1.1
	FMT_SMF	FMT_SMF.1	FMT_SMF.1.1

5.1.1 Security Audit (FAU)

FAU_GEN_MQ.1.1 The TSF shall be able to generate an audit record of the following auditable events:

- Authorization failures.

FAU_GEN_MQ.1.2 The TSF shall record within each audit record at least the following information:

- Date and time of the event, type of event and subject identity; and
- The type of the application causing the event.

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

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to *prevent* unauthorised modifications to the audit records in the audit trail.

5.1.2 Access Control (FDP)

FDP_ACC.1.1 The TSF shall enforce the *access control policy* on *processes acting on behalf of users, objects and all operations among processes acting on behalf of users and the following objects:*

- *Queues;*
- *Process definitions;and*
- *Namelists.*

FDP_ACF.1.1 The TSF shall enforce the *access control policy* to objects based on *the following:*

Subject	*Security Attributes*
Process acting on behalf of a user	*User/Group IDs*
Object	*Security Attributes*
Queues	*ACL*
Process definitions	*ACL*
Namelists	*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 the subject's user or group ID is present within the object's ACL then access is permitted.*

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *no additional rules*.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the: *no additional rules*.

5.1.3 Security Management (FMT)

FMT_MSA.1.1 The TSF shall enforce the *access control policy* to restrict the ability to *modify* the security attributes *ACL* to *the administrator*.

FMT_MSA_MQ.3.1 The TSF shall enforce the access control policy to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MTD.1.1 The TSF shall restrict the ability to *delete* the *event messages* to the *administrator*.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: *Object Security Attributes management*.

5.2 Strength Of Function (SOF)

There is no strength of function claim because the TOE does not identify any security functional requirements for which an explicit Strength of Function (SOF) is appropriate and does not identify any functions that are of a permutational or probabilistic nature. Therefore, a minimum SOF claim is not included for the TOE.

5.3 TOE Security Assurance Requirements

The target evaluation assurance level for this product is EAL2. No augmented assurance requirements are defined.

5.4 Security Requirements for the IT Environment

This section specifies the Security Requirements for the IT environment and organises the requirements by class.

Within the text of each SFR, the selection and assignment operations (as defined within [CC]) are *italicised*.

CLASS	FAMILY	COMPONENT	ELEMENT
FIA	FIA_ATD	FIA_ATD.1	FIA_ATD.1.1

	FIA_UID	FIA_UID.2	FIA_UID.2.1
FMT	FMT_SMR	FMT_SMR.1	FMT_SMR.1.1
			FMT_SMR.1.2
FPT	FPT_STM	FPT_STM.1	FPT_STM.1.1

5.4.1 Identification and Authentication (FIA)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: *User and Group IDs.*

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

5.4.2 Security Management (FMT)

FMT_SMR.1.1 The TSF shall maintain the role *administrator.*

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

5.4.3 Protection of the TSF (FPT)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

6 TOE Summary Specification

6.1 IT Security Functions (SF)

6.1.1 Access Control

AC.1 The TSF shall ensure that access to an object is only given to a process acting on behalf of a user, if the associated user and group IDs associated with the user, has been granted permission to access to that object. The user and group IDs are gained from the operating system and cached in memory for any subsequent access requests. Each process contains the user ID within the message descriptor part of the process, which is used to confirm the group permissions. Permission is confirmed by checking that either the UID or GID is contained within the object's Access Control List (ACL).

AC.2 The TSF shall ensure that only the administrators are able to modify the ACL or delete event messages. Administrators are users that belong to the *MQM* or administrator groups within the Operating System environment. Identification is performed in the same way as normal users.

AC.3 On creation of an object, the TSF shall set default values for that object such that only the ID associated with the process creating the object and the administrator are able to access that object. This is done by adding the creators and administrators UID and GIDs to the ACL of that object. Once an object has been created, then the administrator can update the ACL to grant or revoke access via the command line interface.

6.1.2 Audit

Audit.1 Provided that the event queue is not full, then the TOE shall generate an event message for Authorisation failures. The Queue Manager will put an event message onto the event queue, which behaves in the same manner as all other queues and like other queues has an ACL list, with access only given to the administrator (i.e. members of the MQM group). If the event queue becomes full, then no auditing will take place.

Audit.2 For each event message, the following information is recorded:

- Date and Time;
- Type of event;
- Type of application that caused the event; and
- User identity.

The date and time information is retrieved from the Operating system each time an event message is created. The User ID is gained from the process message descriptor. The Type of event in this case is authorisation failure. Viewing of the audit records is performed via a third party application.

Audit.3 The event queue will be protected to prevent unauthorised modification and deletion of audit records. This is done in the same way as all other queues (see AC.1) with only the administrator (member of MQM group) being able to access the queue. Queue administration is performed using the MQSC commands, which are initialised by entering the *runmqsc* control command at the administrative interface.

6.2 Assurance Measures

Assurance measures will be adopted to address each of the EAL2 assurance requirements, as summarised in table B.1 within [CC] and the International Interpretations 003, 004, 016, 019, 027 and 051 (Rev.1). The following table provides a summary:

Assurance Component	Description of how Requirement will be met
ACM_CAP.2	A description of the configuration management used by the developers will be provided to the evaluators together with a configuration list, which will identify the items that comprise the TOE. This document will uniquely reference the TOE stated within Section 1 of this ETR. Confirmation that the TOE is labelled with the correct reference will be provided during testing.
ADO_DEL.1	The developers will provide the evaluators with the delivery procedures used to ensure that security is maintained when distributing versions of the TOE to the user's site. This is contained within the Configuration Management documentation.
ADO_IGS.1	Procedures for the secure installation, generation and start-up is provided at the following URL: http://www-3.ibm.com/software/integration/mqfamily/library/manualsa/manuals/platspecific.html
ADV_FSP.1	An informal description of the TSF and its external interfaces, describing effects, exceptions and interfaces will be provided to the evaluators.
ADV_HLD.1	A high-level design will be provided to the evaluators, which informally describes the components of the TSF. The security of each of these components will be described. All hardware, software and firmware required by the TOE will be identified. A presentation of the functions provided by the supporting protection mechanisms implemented in these, will also be included. It will also identify the interfaces between the components and which of these are externally visible.
ADV_RCR.1	This correspondence information will be contained within the Functional Specification and high-level design. This will provide a correspondence analysis between the TOE summary specification, the functional specification and the high level design.

AGD_ADM.1	The WebSphere MQ operational documentation that describes to the administrator how to operate the TOE in a secure manner is provided at the following URL: http://publibfp.boulder.ibm.com/epubs/html/amqzag04/amqzag04tfrm.htm This describes the administrative security functions and interfaces available to the administrator. All details of any warnings about functions and privileges and assumptions about user behaviour are included. Secure parameters under the control of the administrator are provided, indicating secure values where applicable.
AGD_USR.1	There are no non-privileged users of the TOE and therefore no documentation shall be provided.
ATE_COV.1	Coverage of the TSF by the developers functional testing to the functional specification will be provided to the evaluators as part of the testing documentation.
ATE_FUN.1	Testing documentation will be provided to the evaluators, which describes the functional tests performed by the developers. This document will include test plans, test procedures, expected and actual test results, It will also identify the security functions to be tested.
ATE_IND.2	Resources will be made available to the evaluators such that they are able to perform additional, independent testing.
AVA_SOF.1	There are no functions within the TOE that have a strength and therefore no Strength of Functions analysis will be produced.
AVA_VLA.1	A description and analysis of any potential vulnerability identified within the TOE will be performed. This will be documented together with an explanation of why the vulnerabilities cannot be exploited.

7 Rationale

This chapter presents the evidence used in the ST evaluation and supports the claims that the ST is a complete and cohesive set of requirements.

7.1 Correlation of Threats, Policies, Assumptions and Objectives

The following table provides a correspondence of the threats, policies, assumptions and objectives:

Objectives:	O.ACCESS	O.ACCOUNT	O.MANAGE	O.IDENTIFY	O.ROLE	O.TIME	O.ADMIN	O.CONFIG	O.OS	O.PROTECT	O.RECOVER
T.ACCESS_RES	x		x							x	x
T.ACCOUNT		x	x			x	x	x			x
T.ACCESS_TOE				x				x			
T.ROLE				x	x			x			
T.NETWORK									x	x	
T.OS							x	x	x	x	x
P.ACCESS	x		x				x	x	x		
A.OS						x	x	x	x		
A.PROTECT										x	
A.ADMIN							x				

7.2 Security Objectives Rationale

This section demonstrates that the security objectives stated in section 4 of this ST are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them.

7.2.1 Threats

This section provides evidence demonstrating coverage of the threats by both the IT and non-IT security objectives.

[T.ACCESS_RES]

An authorised user of the TOE gains access to an object without the correct authority to access that object.

The objective O.ACCESS counters this directly by ensuring that only those users with the correct authority can access an object. This is supported by O.MANAGE, which ensures that privileged actions are performed effectively.

The following environmental objectives support O.ACCESS in countering the threat:

- O.PROTECT – ensures that no objects can be accessed by the cabling between the workstations;
- O.RECOVER – ensures that following a system failure, the TOE is not operating in an insecure state whereby an unauthorised user can gain access to objects they are not authorised to access.

[T.ACCOUNT]

Unauthorised attempts to access objects for which the user has no authority go undetected.

Recording unsuccessful attempts to access objects is performed by O.ACCOUNT. O.MANAGE supports this objective by ensuring that event messaging functions is always invoked and cannot be bypassed.

O.ADMIN, O.CONFIG and O.RECOVER further support these objectives by ensuring that the administrator manages the event messaging security functions effectively. [O.TIME] ensures that the time information recorded for each event is accurate.

[T.ACCESS_TOE]

An unidentified user gains access to the TOE and it's objects.

O.IDENTIFY is the primary objective that counters this threat, by ensuring that all users are identified. The environmental objective O.CONFIG supports O.IDENTIFY in countering the threat by ensuring that all users have a valid and unique identity.

[T.ROLE]

A non-privileged user gains administrative privileges.

Only those users with the correct authority can invoke administrative privileges. O.ROLE ensures that the users are associated with roles so enable the efficient management of administrative users, and maintains an administrative role in order that the TOE can be managed. This relies upon O.IDENTIFY and O.CONFIG, which ensure that each user is identified.

[T.NETWORK]

Data transferred between workstations is disclosed to, or modified by unidentified users or processes, either directly or indirectly.

Administrators must ensure that data transferred between workstations i.e. along network cabling, is suitably protected against physical or other (e.g. Sniffing) attacks that may result in the disclosure, modification or delay of information transmitted between workstations. Objective O.PROTECT ensures that this is achieved. O.OS ensures that the protocols used in the transmission of data have been correctly configured within the operating systems.

[T.OS]

The operating system on which the TOE is installed becomes compromised.

It is essential that the administrator manage the operating system in a secure manner so that vulnerabilities do not exist, which may lead to compromise of the TOE. The objectives O.OS, O.CONFIG, O.PROTECT and O.RECOVER all ensure that the operating system is managed in a secure manner. O.ADMIN further supports this threat by ensuring that the administrator is a competent individual that will apply the latest patch information and therefore ensuring that any vulnerabilities to the TOE that become known are be countered by application of the relevant patch.

7.2.2 Security Policy

This section provides evidence demonstrating coverage of the organisational security policy by both the IT and non-IT security objectives.

[P.ACCESS]

The right to access a specific object is determined on the basis of:

- *The identity of the subject attempting to access the object; or*
- *Membership of a group that has access rights to the object.*

This policy is implemented through the objective O.ACCESS, which provides the means of controlling access to objects by users and processes. O.MANAGE supports this policy by the administrators ensuring that the policy is maintained.

The following environmental objectives further support the policy:

- O.ADMIN, O.CONFIG and O.OS all ensure that the operating system is configured in a secure manner so that no vulnerability may exist that enables an unauthorised user to gain an authorised identity.

7.2.3 Assumptions

This section provides evidence demonstrating coverage of the assumptions by both the IT and non-IT security objectives.

[A.OS]

It is assumed that the operating system has been configured in accordance with the manufacturers instructions and where applicable, the evaluated configuration. It is securely configured such that the operating system protects the TOE from any unauthorised users or processes.

O.OS is the primary environmental objective that satisfies the assumption. This ensures that the administrator installs and configures the supporting operating systems in accordance with:

- The manufacturers instructions; and
- Any evaluated configurations were applicable.

O.ADMIN and O.CONFIG supports this by ensuring that the Administrator is a competent and trustworthy person and that the users have been set up appropriately. [O.TIME] ensures that the time provided by the OS is accurate and reliable.

[A.PROTECT]

It is assumed that all software and hardware, including network and peripheral devices, have been approved for the transmittal of data. Such items are to be physically protected against threats to the confidentiality and integrity of the data.

The environmental objective O.PROTECT ensures that the network cabling is suitably protected against threats of modification, tampering or interruption of the data transmitted via this medium.

[A.ADMIN]

It is assumed that there are one or more competent individuals that are assigned to manage the TOE and the security of the information it contains. Such personnel are assumed not to be careless, wilfully negligent or hostile.

O.ADMIN is the primary objective that meets this assumption, which ensures that the administrator is a competent and trustworthy person whom is capable of managing the TOE in a secure manner.

7.3 Security Requirements Rationale

7.3.1 Security Functional Requirements Rationale

This section demonstrates that the functional components selected for the TOE provide complete coverage of the defined security objectives. The mapping of components to security objectives is illustrated in the table below.

Security Objective	Functional Component
O.ACCESS	Subset Access Control (FDP_ACC.1) Security Attribute Based Access Control (FDP_ACF.1) Management of Security Attributes (FMT_MSA.1) Static Attribute Initialisation (FMT_MSA_MQ.3)
O.ACCOUNT	Audit Data Generation (FAU_GEN_MQ.1) User Identity Association (FAU_GEN.2) Protected Audit Trail Storage (FAU_STG.1) Management of TSF Data (FMT_MTD.1)
O.MANAGE	Management of Security Attributes (FMT_MSA.1) Static Attribute Initialisation (FMT_MSA_MQ.3) Management of TSF Data (FMT_MTD.1) Specification of Management Functions (FMT_SMF.1)

[O.ACCESS]

The TOE must ensure that only those users with the correct authority are able to access an object.

The access control mechanism must have a defined scope of control [FDP_ACC.1] with defined rules [FDP_ACF.1]. Authorised users must be able to control who has access to the objects [FMT_MSA.1]. Protection of these objects must be continuous, starting from object creation [FMT_MSA_MQ.3]

[O.ACCOUNT]

The TOE must provide a means of recording any unsuccessful access attempts to the objects.

Security relevant actions must be defined, auditable [FAU_GEN_MQ.1] and capable of being associated with individual users [FAU_GEN.2]. The event queue must be protected

so that only authorised users can access it [FAU_STG.1]. An authorised administrator must be able to manage the event queue [FMT_MTD.1].

[O.MANAGE]

The TOE must allow administrators to effectively manage the TOE and that this is only performed by authorised users.

The TSF must enable an authorised administrator to manage the TOE by the access control policy objects [FMT_MSA.1] with default values [FMT_MSA_MQ.3]. The administrator must be able to manage the event queue [FMT_MTD.1] and the access control list [FMT_MSA.1]. [FMT_SMF.1] specifies the management functions provided by the TOE.

7.3.2 Security Environment Requirements Rationale

This section demonstrates that the functional components provided by the environment for the TOE, provide complete coverage of the defined security objectives. The mapping of requirements to security objectives is illustrated in the table below.

Security Objective	Requirement for Environment
O.IDENTIFY	User Attribute Definition (FIA_ATD.1) User Identification (FIA_UID.2)
O.ROLE	User Identification (FIA_UID.2) Security Management (FMT_SMR.1)
O.TIME	Protection of the TSF (FPT_STM.1)

[O.IDENTIFY]

The Operating System must ensure that all users are identified.

The TSF must maintain a list of User and Group IDs for each user (FIA_ATD.1) and identify users before allowing any other actions (FIA_UID.2).

[O.ROLE]

The operating system must be able to associate users with roles and maintain an administrator role.

In order to associate a user with a role (FMT_SMR.1), the user needs to be identified (FIA_UID.2) and maintain administrative roles (FMT_SMR.1).

[O.TIME]

The operating system must ensure that the clock is accurate and reliable.

In order that the TOE is able to provide accurate time stamps, in this case for audit records, the operating system that the TOE is relying for the time information must ensure that this is reliable and accurate (FPT_STM.1).

7.3.3 Explicitly Stated Security Requirements Rationale

As stated within Section 5 of this ST, FAU_GEN_MQ.1 and FMT_MSA_MQ.3 have been explicitly stated and were not specified using CC Part 2 functional components. The reasons for this are as follows:

FAU_GEN_MQ.1

The TOE does not generate an audit record for the start-up and shutdown of the auditing functions and the success/failure of the events audited is not recorded.

These do not lead to any vulnerability within the system because only the administrator is able to start-up and shutdown the auditing functions and is trusted to operate the system securely. If an unauthorised user were able to start-up and shutdown the auditing function, then that user would have administrative rights and would therefore be capable of performing any action on the TOE.

Auditing of every successful attempt to access an object would create an impractically large audit file with no benefit to the administrator. Therefore only *unsuccessful* attempts to access an object generate an audit record.

FMT_MSA_MQ.3

WMQ does not provide functionality to define alternate initial values that override the default values when an object has been created. This does not reduce security as the default values used are the most restricted that would enable normal operation of the TOE.

7.3.4 Security Assurance Requirements Rationale

This ST contains assurance requirements from the CC EAL2 assurance package.

The EAL chosen is based on the impact that the statements of the security environment and objectives within this ST have on the assurance level. The administrator shall be capable of managing the TOE such that the security is maintained (O.ADMIN) particularly within the operating system that the TOE relies (O.OS), and that the physical environment protects the TOE from any potential vulnerability (O.PROTECT). This EAL level also provides a low to moderate level of independently assured security without demanding additional effort by the developers.

Given the amount of assurance required to meet the TOE environment and the intent of EAL2, this assurance level was considered most applicable for the TOE described within this ST.

7.4 SFR Dependencies

The below table identifies all of the dependencies of the SFRs included in the ST. Only those SFRs that have a dependency, or are depended upon are shown in the table.

	FAU_GEN.1*	FDP_ACC.1	FDP_ACF.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.3*	FMT_SMF.1	FMT_SMR.1	FPT_STM.1
FAU_GEN_MQ.1									x
FAU_GEN.2	x			x					
FAU_STG.1	x								
FDP_ACC.1			x						
FDP_ACF.1		x				x			
FMT_MSA.1		o					x	x	
FMT_MSA_MQ.3					x			x	
FMT_MTD.1							x	x	

The key to the symbols used, are:

x required dependency

o optional dependency

As shown in [CC], all dependencies are satisfied by the TOE, with the exception of the dependencies on FIA_UID.1, FMT_SMR.1 and FPT_STM.1. These dependencies are met by the IT environment of the TOE.

FIA_UID.1 is countered by the IT environment of the TOE because the Operating system provides the TOE with the user IDs (FIA_UID.2). Additionally, this is met by FIA_UID.2 as it is hierarchical to FIA_UID.1.

The TOE does not 'maintain' an administrator role (FMT_SMR.1) or internal clock (FPT_STM.1), but relies upon the operating system to maintain the role and clock. Identification of the administrator is based on membership of the mqm or administrators group defined within the operating system. [A.OS] assumes that *the operating system has been configured in accordance with the manufacturer's installation guides and where applicable, in its evaluated configuration. It is securely configured such that the operating system protects the TOE from any unauthorised users or processes*.

* The reliance on the requirements FAU_GEN.1 and FMT_MSA.3 are countered by the explicitly stated requirements FAU_GEN_MQ.1 and FMT_MSA_MQ.3 respectively, which records auditing details and sets default values on creation of objects. In turn, the

explicitly stated requirements assume the dependencies on FAU_GEN.1 and FMT_MSA.3 as shown in the above table.

7.5 TOE Summary Specification Rationale

This section demonstrates that the TOE security functions and assurance measures are suitable to meet the TOE security requirements.

7.5.1 TSF correspondence to SFRs

This section demonstrates that the combination of the specified TSFs work together so that the SFRs are satisfied. The table below shows the TOE security functions, which together satisfy each SFR element.

TOE SFR	**TSFs**
FAU_GEN_MQ.1.1	Audit.1
FAU_GEN_MQ.1.2	Audit.2
FAU_GEN.2.1	Audit.2
FAU_STG.1.1	Audit.3
FAU_STG.1.2	Audit.3
FDP_ACC.1.1	AC.1
FDP_ACF.1.1	AC.1
FDP_ACF.1.2	AC.1
FDP_ACF.1.3	AC.1
FDP_ACF.1.4	AC.1
FMT_MSA.1.1	AC.2
FMT_MSA_MQ.3.1	AC.3
FMT_MTD.1.1	AC.2
FMT_SMF.1.1	AC.1, AC.2 and AC.3

