MarkLogic Server Enterprise Edition Version 4.0

Security Target

Version 1.0

June 29, 2010

Prepared for:
Mark Logic Corporation
999 Skyway Road, Suite 200
San Carlos, CA 94070

Prepared By:
Science Applications International Corporation
Common Criteria Testing Laboratory
7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

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 MarkLogic Server Enterprise Edition Version 4.0 provided by Mark Logic Corporation. MarkLogic Server Enterprise Edition is an enterprise-class database or "contentbase" that provides a set of services used to build both content and search applications which query, manipulate and render XML content.

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)
- Rationale (Section 8)
- Appendix (Section 9).

1.1 Security Target, TOE and CC Identification

ST Title – MarkLogic Server Enterprise Edition Version 4.0

ST Version – Version 1.0

ST Date – June 29, 2010

TOE Identification – MarkLogic Server Enterprise Edition Version 4.0

TOE Developer – Mark Logic Corporation

Evaluation Sponsor – Mark Logic Corporation

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007.

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 3.1, Revision 2, September 2007.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 3.1, Revision 2, September 2007.
    - Part 3 Conformant
    - Assurance Level: EAL 3 augmented with ALC_FLR.3
- The TOE is further conformant to the following Protection Profile (PP):

- U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2, July 25, 2007 (hereafter referred to as the DBMS PP).

1.3 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**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-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 …").
  - Extended security functional requirements are indicated with "EXT".
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Acronyms

API	Application Programming Interface
CC	Common Criteria
CEM	Common Evaluation Methodology
CCEVS	Common Criteria Evaluation and Validation Scheme
CIM	Consistency Instruction Manual for Development of U.S. Government Protection Profiles for use in Basic Robustness Environments
DAC	Discretionary Access Control
DBMS	Database Management System
DBMS PP	U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2, July, 25, 2007
DoD	Department of Defense
DoS	Denial of Service
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
HLD	High-level Design
IA	Initial Assessment
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OS	Operating System
PP	Protection Profile
SAIC	Science Applications International Corporation

SFP	Security Function Policy
SOF	Strength of Function
SQL	Structured Query Language
ST	Security Target
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Functions
URI	Uniform Resource Identifier
US	United States
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1.3.2 Terminology

The terminology below is described in order to clarify and distinguish the terms used in the Protection Profile, the ST and those used in the TOE product documentation.

Group

The DBMS PP specifies that the discretionary access control policy (DAC) is based on a user's identity and/or group membership. The term "group" as used in the DBMS PP is equivalent to the concept of "role" which is the terminology used by MarkLogic.[1] Therefore, for purposes of this ST and consistency with DBMS PP terminology, the terms "group(s)" is used, but refers to the concept of "role" that is described in other TOE documentation and is defined below.

Role

In the DBMS PP, the term "role" is used to refer to the security relevant database roles that are defined for the TOE. For the MarkLogic TOE, one security relevant role, authorized administrator, has been identified. The Marklogic TOE actually implements and enforces other user roles, however, these translate into "groups" and are discussed as such for purposes of this ST and consistency with the DBMS terminology. A role (i.e. group) is a named entity that provides authorization privileges and permissions to other roles (i.e. groups) or to users. Users, privileges, document permissions and other roles (i.e. groups) are all assigned to roles which are "groups" in this ST.

Note (1): Apart from the authorized administrator role defined in this ST, MarkLogic TOE user roles, shall, henceforth be referred to as groups.

Note (2): The following roles are pre-defined in a MarkLogic Server installation: admin, admin-builtins, alert-admin, alert-internal, alert-user, domain-management, filesystem-access, merge, pipeline-management, security and trigger-management. With the exception of the admin role, none of these pre-defined roles are included in the evaluated configuration of the TOE. The admin role is the TOE's authorized administrator security role and any other roles used must be created by an authorized administrator using the Admin Interface.

[1] In MarkLogic product documentation, the term "group" actually refers to a set of similarly configured hosts in a cluster. This ST refers to these groups as "Cluster Management Groups" so as to distinguish them from the user groups in the ST.

Amps	Amps are security objects that temporarily grant group membership to unprivileged users only for the execution of a given function. While executing an "amped" function, the user is temporarily part of the amped group which in turn temporarily grants the user the additional privileges and permissions given by the groups configured in the amp. Amps enable the effect of the additional permissions and privileges to be limited to a particular function.
Permissions	Permissions provide a group with the ability to perform certain capabilities (i.e. read, insert, update, execute) on documents. Permissions are assigned to documents. Users gain the authority to perform these capabilities on a document if they are members of a group to which a permission is associated.
Capabilities	Permissions are a combination of group and a capability. Capabilities are: Read, Update, Insert or Execute.
Execute Privileges	Execute privileges allow developers to control authorization for the execution of an XQuery function. These privileges are assigned to a user through a group.
URI Privileges	Uniform Resource Identifier privileges are used to control the creation of documents with a given URI prefix. In order to create a document with a prefix that has a URI privilege associated with it, a user must be part of a group to which the needed URI privilege is assigned.
Application Server Privileges	Application Server Privileges are Execute Privileges that can be configured to control access to each application server (i.e. HTTP or XDBC server). If such a privilege is specified, any users that access the server must possess the specified privilege.

2. TOE Description

The Target of Evaluation (TOE) is MarkLogic Server Enterprise Edition, Version 4.0, hereafter referred to as MarkLogic Server or the TOE.

2.1 TOE Overview

The TOE is Mark Logic Corporation's MarkLogic Server software. MarkLogic Server is an enterprise-class database or "contentbase" that provides a set of services used to build both content and search applications which query, manipulate and render Extensible MarkUp Language (XML) content.

The MarkLogic Server TOE is built with a blend of search engine and database architecture approaches specifically designed to index and retrieve XML content. The TOE's native data format is XML and XML is accepted in an 'as is' form, while content in other formats can be converted to an XML representation or stored as is (in binary or text formats) when loaded into the server. As an XML content server, it manages its own content repository and is accessed using the W3C standard XQuery language, just as a relational database is a specialized server that manages its own repository and is accessed through Structured Query Language (SQL).

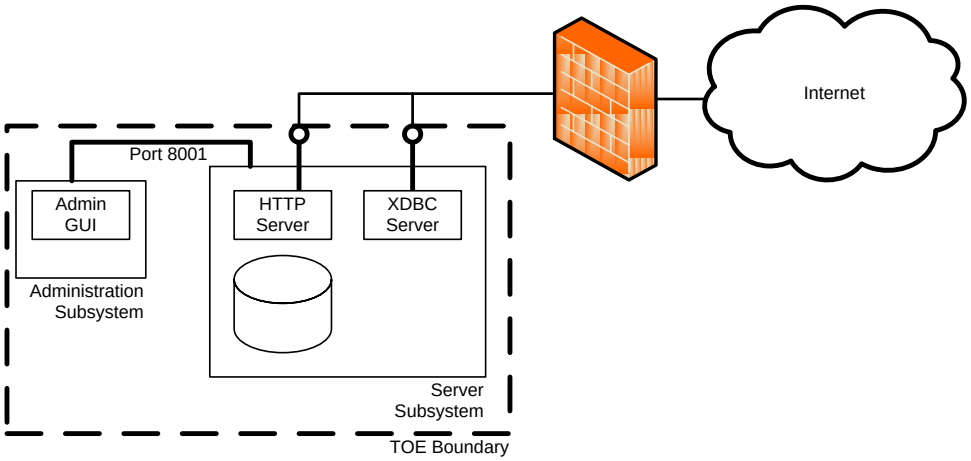
The TOE is fully transactional, runs in a distributed environment and can scale to terabytes of indexed content. It is schema independent and all loaded documents can be immediately queried without normalizing the data in advance. Like a relational database, it provides developers with the functionality and programmability, using XQuery as its query language, to build content-centric applications. Developers build applications using XQuery both to search the content and as a programming language in which to develop applications. It is possible to create entire applications using only MarkLogic Server, and programmed entirely in XQuery.

The security management functions of the TOE are performed via the Admin Interface,, which is a web based browser GUI implemented as a MarkLogic Server web application. This interface allows authorized administrators to manage audit events, user accounts, access control and TOE sessions. It also provides the ability to control the creation, management, and configuration of databases, forests, servers, and hosts. Documents are stored in forests. The name forests comes from the fact that XML documents are tree structures and a collection of trees is a forest. One or more forests are gathered together to form a database. Databases are logical units against which you can assign HTTP and XDBC servers and set various runtime configuration options. A host is a single instance of MarkLogic Server running on a single machine. Databases exist as a logical abstraction because in a distributed environment it can be useful to have the same logical database spread across different hosts, perhaps one host with two forests and another with three.

2.2 TOE Architecture

The TOE consists of two subsystems, the Administration subsystem and the Server subsystem. The Administration subsystem provides the Admin Interface to the Server subsystem. The Admin Interface application manages all features of the Server subsystem. It is composed of XQuery programs which are evaluated inside of an HTTP server. The HTTP server evaluates each request and sends a response back as a web page to the requester. The Admin Interface runs on Port 8001 behind a firewall which is configured to block egress and ingress of traffic over Port 8001.

The Server subsystem provides the software applications, network/application programming interfaces (APIs) and a database or contentbase as shown in the TOE architecture diagram below:



TOE Architecture

The network/programmatic interfaces (HTTP and XDBC) are used by developers in a system that requires access to a backend XML content store.

The TOE can be set up as a single instance of MarkLogic Server on a single machine or it can support large scale high-performance architectures through multi-host distributed architectures. The following terminology has been defined for consideration in a TOE distributed environment:

- Cluster – A cluster is a set of one or more instances (see hosts, below) of MarkLogic Server (i.e. the TOE's Server subsystem) that will work together as a unified whole to provide content services. Security management functions of the TOE are performed from the Administration subsystem by connecting to any cluster host.

- Host – A host is a single instance of MarkLogic Server running on a single machine. Even though each host in a cluster can be configured to perform a different task, the full MarkLogic Server software (Server subsystem) runs on each host. MarkLogic Server Standard Edition can only be configured to run in a single-host configuration. MarkLogic Server Enterprise Edition enables multi-host configurations.[2]

- Cluster Management Group – A cluster management group is a set of hosts with uniform HTTP and XDBC server configurations (but not necessarily uniform forest configurations). Cluster Management Groups are used to simplify cluster management.

- Forest – A forest is a repository for documents. Each forest is managed by a single host. The mapping of which forest is managed by which host is transparent to queries, as queries are processed against databases, not forests.

- Database – A database is a set of one or more forests that appears as a single contiguous set of content for query purposes. Each forest in a database must be configured consistently. HTTP and XDBC servers evaluate queries against a single database. In addition to databases created by the administrator for user content, MarkLogic Server maintains databases for administrative purposes: *security* databases, which contain user authentication and permissions information; *schema* databases, which are used to store schemas used by the system; *modules* databases, which are used

[2] The evaluated configuration only includes the MarkLogic Server Enterprise Edition which supports multi-host configurations.

to store executable XQuery code; *last-login* databases, which are used to store session history and data and *triggers* databases, used to store trigger definitions.

2.2.1 Physical Boundaries

The TOE consists of the software applications and network protocol interfaces (described and shown in the diagram above). The Administration subsystem, which provides the Admin Interface, runs on Windows XP SP2 using Internet Explorer v.6.0 or higher. The Server subsystem applications and network interfaces execute either on Sun Solaris or Linux operating systems. The TOE requires the following hardware and operating system (OS) platforms in the IT environment:

Memory, Disk Space, and Swap Space Requirements
Before installing the software, the system must meet the following minimum requirements:

- 512 MB of system memory, minimum.
- Three times the disk space of the source content to be loaded.
- Swap space at least equal to the amount of physical memory on the machine.

Supported Platforms – Server Subsystem
The MarkLogic Server server subsystem is supported on the following platforms for the evaluated configuration:

- Sun Solaris 10 (64-bit SPARC)
- Sun Solaris 10 (x64)
- Red Hat Enterprise Linux 5.0 (x64)

Supported Platforms – Administration Subsystem
The MarkLogic Server administration subsystem is supported on the following platforms for the evaluated configuration:

- Microsoft Windows XP SP2.

As noted previously, the TOE can be deployed on a single machine or in a distributed environment across multiple machines.

The TOE relies on the hosting OS to protect its applications, processes, and any locally stored data. Web browsers in the IT environment are utilized to access the Admin Interface and the HTTP server, and to terminate a session. The Admin Interface prompts the user to authenticate with a valid username and password in order to log in for a session. As is standard in browser-based applications, the browser caches and automatically re-issues the login credentials for each request throughout the browser session. These credentials are valid until the browser is closed, which terminates the session. When the browser is restarted, the user will once again be prompted to authenticate with a valid username and password.

2.2.2 Logical Boundaries

This section identifies the security functions that MarkLogic Server Enterprise Edition, Version 4.0 provides**.** The logical boundaries of the TOE include the security functions of the TOE interfaces. The TOE logically supports the following security functions:

- Security Audit
- Identification and Authentication
- Security Management

- User Data Protection
- Protection of the TSF
- TOE Access

2.2.2.1 Security audit

The TOE generates audit records that include date and time of the event, subject identity and outcome for security events. The TOE provides authorized administrators with the ability to include and exclude auditable events based on group identity, event type, object identity and success and failure of auditable security events. When appropriate, the TOE also associates audit events with the identity of the user that caused the event. The IT environment stores the audit records and also provides the system clock information that is used by the TOE to timestamp each audit record.

2.2.2.2 User data protection

The TOE enforces a Discretionary Access Control (DAC) policy which restricts access to DBMS-controlled object(s). Users of the TOE are identified and authenticated by the TOE before any access to the system is granted. Once access to the system is granted, authorization provides the mechanism to control what functions a user is allowed to perform based on the user's group membership. Access to all DBMS-controlled objects is denied unless access, based on group membership, is explicitly allowed. The authorized administrator role shall be able to bypass the DAC policy. The TOE also provides amplifications or "amps" which temporarily grant roles to a user only for the execution of a specific function. Therefore, the DAC policy can also be bypassed by a user who is temporarily granted the authorized administrator role in order to perform a specific "amped" function. The TOE also ensures that any previous information content of a resource is made unavailable upon the allocation of the resource to an object. Memory or disk space is only allocated when the size of the new data is first known, so that all previous data is overwritten by the new data.

2.2.2.3 Identification and authentication

The TOE requires users to provide unique identification and authentication data before any access to the system is granted and further restricts access to DBMS-controlled objects based on group membership. The TOE maintains the following security attributes belonging to individual users: group membership, security-relevant database role and password. The TOE uses these attributes to determine access.

2.2.2.4 Security management

The security functions of the TOE are managed by authorized administrators via the web based Admin Interface. The TOE defines the security role of 'authorized administrator.' Authorized administrators perform all security functions of the TOE including managing audit events, user accounts, access control and TOE sessions.

2.2.2.5 Protection of the TSF

The TOE provides protection mechanisms for its security functions. One of the protection mechanisms is that users must authenticate and have the appropriate permissions before any administrative operations or access to TOE data and resources can be performed on the system. The TOE also maintains a security domain that protects it from interference and tampering by untrusted subjects within the TOE scope of control. Additionally, the TOE ensures that TSF data is consistent between parts of the TOE with a mechanism that brings inconsistent data into a consistent state.

2.2.2.6 TOE access

The TOE restricts the maximum number of concurrent sessions that belong to the same user by enforcing an administrator configurable number of sessions per user. The TOE also denies session establishment

based on attributes that can be set explicitly by authorized administrators including group identity, time of day and day of week. Upon successful session establishment, the TOE stores and retrieves the date and time of the last successful session establishment to the user. It also stores and retrieves the date and time of the last unsuccessful session establishment and the number of unsuccessful attempts since the last successful session establishment.

2.3 TOE Documentation

Mark Logic has a number of administration and configuration guides for the TOE which include the following:

- MarkLogic Server Administrator's Guide, Release 4, September 2008
- MarkLogic Server Understanding and Using Security, Release 4.0, September 2008
- MarkLogic Server Scalability, Availability, and Forest-Level Failover, Release 4.0 September, 2008, Last Revised: 4.0-1, September, 2008
- MarkLogic Server Developer's Guide, Release 4.0, September 2008
- MarkLogic Server Enterprise Edition 4.0 Security Architecture, August 2009
- MarkLogic Common Criteria Evaluated Configuration Guide, Release 4.0 TOE Draft, July 2009
- MarkLogic Server Installation Guide for All Platforms, Release 4.0, September, 2008

3. Security Environment

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

- Threats that the product is designed to counter.
- Assumptions made on the operational environment and the method of use intended for the product.
- Organizational Policies with which the product is designed to comply.

The assumptions, threats and organizational security policies are taken directly from the DBMS PP. With the exception of two additional threats, T.PRIV and T.OPS, and three additional assumptions, A.ADMIN, A.AUTH and A.BROWSER there are no modifications to the security environment of the PP.

3.1 Organizational Policies[3]

P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.ROLES	The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

3.2 Threats

T.ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.OPS	An unauthorized process or application may gain access to the TOE security functions and data, inappropriately changing the configuration data for the TOE security functions.
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that

[3] In compliance with the DBMS PP, the Organizational Policies recommended by the CIM that do not apply to the TOE have been included in this Security Target in the Appendix, Section 9.1.

	may be exploited by a casually mischievous user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.
T.PRIV	An unauthorized user may gain access to the TOE and exploit system privileges to gain access to TOE security functions and data, inappropriately changing the configuration data for TOE security functions.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.TSF_COMPROMISE	A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

3.3 Assumptions

A.ADMIN	The Admin Interface application runs on Port 8001 behind a firewall which is configured to block egress and ingress of traffic over Port 8001.
A.AUTH	Passwords are encrypted during the authentication process.
A.BROWSER	The web browsers used to access the Admin Interface perform correctly such that when the browser is closed, the active Admin session is terminated.
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.

A.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.
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.

4. Security Objectives

The section defines the security objectives of the TOE and its supporting environment. Security objectives for the TOE reflect the stated intent to counter identified threats. All of the identified threats are addressed under one of the categories below.

The security objectives are taken directly from the DBMS PP. With the exception of two additional security objectives for the TOE, O.ACCESS and O.PROTECT, and two additional security objectives for the TOE environment, OE.AUTH and OE.BROWSER, there are no modifications to the security environment of the PP.

4.1 Security Objectives for the TOE

O.ACCESS	The TOE must allow only authorized users and processes (applications) to access protected TOE functions and data.
O.ACCESS_HISTORY	The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide authorized administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
O.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.

O.PARTIAL_FUNCTIONAL_TEST	The TOE will undergo some security functional testing that demonstrates that the TSF satisfies some of its security functional requirements.
O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.
O.PROTECT	The TOE must protect its functions and data from unauthorized access and modifications.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.

4.2 Security Objectives for the Environment

OE.AUTH	Password encryption during the authentication process is provided by the web browser.
OE.BROWSER	The web browsers used to access the Admin Interface will perform correctly and when the browser is closed, the active Admin session will be terminated.
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.NO_GENERAL_ PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.

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.

5. IT Security Requirements

This section provides a list of all security functional requirements (SFRs) for the TOE. The Security functional requirements in this ST are drawn directly from the DBMS PP and consist of SFRs reproduced and/or refined from the CC v3.1, Part 2 as well as some extended requirements. Although, the DBMS PP has been updated to CCv3.1, the minor modifications to the SFRs from CC v2.3 to CCv3.1were not addressed by the editor of the document who wished to ensure that the author's original intent was preserved. The ST author has made modifications to update the SFRs to 3.1 where it was clear that the essential meaning of the requirement was not changed. These changes have been noted in the Application Notes following the associated requirements.

With the exception of the two modifications described below, these requirements have been reproduced with only minor changes made in accordance with CC v3.1. References to table heading numbers and section heading numbers within the requirement statements have been changed as sections and tables in the ST do not have the same exact heading numbers as in the PP. The Auditable Events table under FAU_GEN.1.2 has been refined to identify the security functional requirements actually included in the ST. All operations have been completed on the requirements in compliance with the PP as indicated using bold and bold-italic text in Section 5.1.

The following modifications have been made to the IT Security Requirements drawn from the DBMS PP:

- The FIT_PPC_EXT.1 (IT Environment Protection Profile Compliance) extended IT Environment Security Functional Requirement has been removed.
- FIA_UID.2 and FIA_UAU.2 have been added to this section and all relevant corresponding sections in the ST.

For further information and rationale regarding these modifications, please refer to Section 8.2.1.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are to be satisfied by MarkLogic Server, Version 4.0.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_GEN_EXT.2: User and/or group identity association
	FAU_SEL.1: Selective audit
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
	FDP_RIP.1: Subset residual information protection
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
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1: Management of security attributes
	FMT_MSA_EXT.3: Static attribute initialization
	FMT_MTD.1: Management of TSF data
	FMT_REV.1a: Revocation
	FMT_REV.1b: Revocation

Requirement Class	Requirement Component
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_TRC_EXT.1: Internal TSF consistency
FTA: TOE access	FTA_MCS.1: Basic limitation on multiple concurrent sessions
	FTA_TAH_EXT.1: TOE access history
	FTA_TSE.1: TOE session establishment

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.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 [***minimum***] level of audit **listed in the table below;** c) **[Start-up and shutdown of the DBMS; d) Use of special permissions (e.g., those often used by authorized administrators to circumvent access control policies); and** e) [***no additional events***]**].**

FAU_GEN.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 (if applicable), and the outcome (success or 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, [**information specified in column three of the table below**].

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1: Audit data generation	None	None
FAU_GEN_EXT.2: User and/or group identity association	None	None
FAU_SEL.1: Selective audit	All modifications to the audit configuration that occur while the audit collection functions are operating.	The identity of the authorized administrator that made the change to the audit configuration.
FDP_ACC.1: Subset access control	None	None
FDP_ACF.1: Security attribute based access control	Successful requests to perform an operation on an object covered by the SFP.	The identity of the subject performing the operation.
FDP_RIP.1: Subset residual information protection	None	None
FIA_ATD.1: User attribute definition	None	None
FIA_UAU.2: User authentication before any action	Unsuccessful use of the authentication mechanism	None
FIA_UID.2: User identification before any action	Unsuccessful use of the user identification mechanism including the user identity provided	None
FMT_MOF.1: Management of security functions behaviour	None	None

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FMT_MSA.1: Management of security attributes	None	None
FMT_MSA_EXT.3: Static attribute initialization	None	None
FMT_MTD.1: Management of TSF data	None	None
FMT_REV.1a: Revocation	Unsuccessful revocation of security attributes.	Identity of individual attempting to revoke security attributes.
FMT_REV.1b: Revocation	Unsuccessful revocation of security attributes.	Identity of individual attempting to revoke security attributes.
FMT_SMF.1: Specification of Management Functions	Use of the management functions.	Identity of the administrator performing these functions.
FMT_SMR.1: Security roles	Modifications to the group of users that are part of a role.	Identity of authorized administrator modifying the role definition.
FPT_TRC_EXT.1: Internal TSF consistency	Restoring consistency.	None
FTA_MCS.1: Basic limitation on multiple concurrent sessions	Rejection of a new session based on the limitation of multiple concurrent sessions.	None
FTA_TAH_EXT.1: TOE access history	None	None
FTA_TSE.1: TOE session establishment	Denial of a session establishment due to the session establishment mechanism.	Identity of the individual attempting to establish the session.

Application Note: In FAU_GEN.1, the DBMS PP refers to the audited events in "table 8", whereas the ST refers to the "table below" as it has not been designated as table 8.

5.1.1.2 User and/or group identity association (FAU_GEN_EXT.2)

FAU_GEN_EXT.2.1 For audit events resulting from actions of identified users and/or identified groups, the TSF shall be able to associate each auditable event with the identity of the user and/or group that caused the event.

5.1.1.3 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall ~~be able to~~ **allow only the administrator** to include or exclude auditable events from the set of audited events based on the following attributes: [**a)** ***user identity and/or group identity, b) event type, c) object identity***], d) [***none***], [**e) success of auditable security events; f) failure of auditable security events; and** [***no additional criteria***]].

5.1.2 User data protection (FDP)

5.1.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [**Discretionary Access Control policy**] on [**all subjects, all DBMS-controlled objects and all operations among them**].

5.1.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**Discretionary Access Control policy**] to objects based on the following: [**the authorized user identity and/or group membership associated with a**

subject; access operations implemented for DBMS-controlled objects; and object identity].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and **DBMS**-controlled objects is allowed:

- **The Discretionary Access Control policy mechanism shall, either by explicit authorized user/group action or by default, provide that database management system controlled objects are protected from unauthorized access according to the following ordered rules[4]:**

[***a) If the requested mode of access is denied to that authorized user, deny access;***
b) If the requested mode of access is permitted to that authorized user, permit access;
c) If the requested mode of access is denied to every group of which the authorized user is a member, deny access;
d) If the requested mode of access is permitted to any group of which the authorized user is a member, grant access;
e) Else, deny access.
].

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

- ***The authorized administrator role shall be able to bypass the DAC policy.***
- ***Amps[5] can be used to temporarily grant the authorized administrator role to an unprivileged user, thereby allowing them to bypass the DAC policy while performing specific functions.***]].

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

Application Note: Regarding FDP_ACF.1.3, the PP states that it allows for the addition of rules that allow administrators to bypass the access control policy.

5.1.2.3 Full residual information protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] [**documents**].

5.1.3 Identification and authentication (FIA)

5.1.3.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**Database user identifier and/or group memberships; Security-relevant database roles; and password**].

---

[4] The first two rules a) and b) do not apply to the TOE because object access is based on group membership rather than user identity. Please refer to OD271 which states: "The 2nd app note under the FDP_ACF.1.2-NIAP-0407 element will be replaced with:

> Application Note: It is not required for the TOE to implement access control based on both user IDs and group IDs. If the TOE only implements access control based on user IDs, the rules containing "any group" or "every group" do not apply. If the TOE only implements access control based on group IDs, the rules containing "to that authorized user" do not apply."

[5] For further information on amplifications or "amps", please refer to Section 6.1.2.

5.1.3.2 User authentication before any action (FIA_UAU.2)

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

5.1.3.3 User identification before any action (FIA_UID.2)

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

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to [***disable and enable***] the functions [**relating to the specification of events to be audited**] to [**authorized administrators**].

5.1.4.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Discretionary Access Control policy**] to restrict the ability to [***[manage]***] **all** the security attributes to [**authorized administrators**].

5.1.4.3 Static attribute initialization (FMT_MSA_EXT.3)

FMT_MSA_EXT.3.1 The TSF shall enforce the [**Discretionary Access Control policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

5.1.4.4 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [***[include or exclude]***] the [**auditable events**] to [**authorized administrators**].

5.1.4.5 Revocation (FMT_REV.1a)

FMT_REV.1a.1 The TSF shall restrict the ability to revoke **[group membership, password, security-relevant database role]** associated with the [***users***] ~~within the TSC~~ **under the control of the TSF** to [**the authorized administrator**].

FMT_REV.1a.2 The TSF shall enforce the rules [

- **On the revocation host, revocation is effective on the next session that starts after the revocation request is committed.**
- **On other hosts in a cluster, revocation is effective no later than the receipt of the next heartbeat received from the revocation host**].

Application Note: This requirement has been modified in accordance with CC v3.1, Revision 2 to include the assignment of a list of security attributes, rather than just stating "security attributes". This does not change the meaning of the requirement or reduce its scope. Additionally, the wording has been slightly modified to retain the exact wording of this requirement from the CC v3.1. The wording in CC v3.1 indicates "under the control of the TSF" which is equivalent in meaning to the words "within the TSC".

5.1.4.6 Revocation (FMT_REV.1b)

FMT_REV.1b.1 The TSF shall restrict the ability to revoke **[access operations]** associated with the [***objects***] ~~within the TSC~~ **under the control of the TSF** to [**the authorized administrator and database users as allowed by the Discretionary Access Control policy**].

FMT_REV.1b.2 The TSF shall enforce the rules [

- **On the revocation host, revocation is effective on the next session that starts after the revocation request is committed.**

- **On other hosts in a cluster, revocation is effective no later than the receipt of the next heartbeat received from the revocation host**.]

Application Note: This requirement has been modified in accordance with CC v3.1, Revision 2 to include the assignment of a list of security attributes, rather than just stating "security attributes". This does not change the meaning of the requirement or reduce its scope. Additionally, the wording has been slightly modified to retain the exact wording of this requirement from the CC v3.1. The wording in CC v3.1 indicates "under the control of the TSF" which is equivalent in meaning to the words "within the TSC".

5.1.4.7 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [
- **manage audit events,**
- **manage user accounts,**
- **manage access control, and**
- **manage TOE sessions**].

5.1.4.8 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**authorized administrator**]; **and [no other roles].**
FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Internal TSF consistency (FPT_TRC_EXT.1)

FPT_TRC_EXT.1.1 The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.

5.1.6 TOE access (FTA)

5.1.6.1 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.
FTA_MCS.1.2 The TSF shall enforce, by default, a limit of [***an admin configurable number of***] sessions per user.

5.1.6.2 TOE access history (FTA_TAH_EXT.1)

FTA_TAH_EXT.1.1 Upon successful session establishment, the TSF shall store and retrieve the [***date and time***] of the last successful session establishment to the user.
FTA_TAH_EXT.1.2 Upon successful session establishment, the TSF shall store and retrieve the [***date and time***] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

5.1.6.3 TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [**attributes that can be set explicitly by authorized administrator(s), including user identity and/or group identity, time of day, day of the week**], **and [application server privilege].**

5.2 TOE Security Assurance Requirements

The security assurance requirements (SARs) in the DBMS PP are the CC v3.1 Part 3 requirements for EAL 2 augmented with ALC_FLR.2. The assurance requirements of this ST are the CC v3.1, Release 2, Part 3 EAL 3 requirements augmented with ALC_FLR.3. The EALs serve as a basis of equivalence between the CC versions, therefore, the assurance requirements in this ST are conformant to the DBMS PP, in that they meet or exceed those requirements.

NOTE: In section 5.3, the PP retains wording indicating that the assurance requirements are from CCv2.1, even though they have been updated to the assurance requirements from CC v3.1.

For further information and rationale, please refer to Section 8.2.2.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.3: Functional specification with complete summary
	ADV_TDS.2: Architectural design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.3: Authorization controls
	ALC_CMS.3: Implementation representation CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_FLR.3: Systematic flaw remediation
	ALC_LCD.1: Developer defined life-cycle model
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: basic design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

Table 2 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Security architecture description (ADV_ARC.1)

ADV_ARC.1.1d The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2d The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3d The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1c The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2c The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3c The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4c The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5c The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.2 Functional specification with complete summary (ADV_FSP.3)

ADV_FSP.3.1d The developer shall provide a functional specification.
ADV_FSP.3.2d The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.3.1c The functional specification shall completely represent the TSF.
ADV_FSP.3.2c The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.3.3c The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.3.4c For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.3.5c For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions associated with invocation of the TSFI.
ADV_FSP.3.6c The functional specification shall summarize the SFR-supporting and SFR-non-interfering actions associated with each TSFI.
ADV_FSP.3.7c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.3.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.1.3 Architectural design (ADV_TDS.2)

ADV_TDS.2.1d The developer shall provide the design of the TOE.
ADV_TDS.2.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.2.1c The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.2.2c The design shall identify all subsystems of the TSF.
ADV_TDS.2.3c The design shall describe the behavior of each SFR non-interfering subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.
ADV_TDS.2.4c The design shall describe the SFR-enforcing behavior of the SFR-enforcing subsystems.
ADV_TDS.2.5c The design shall summarize the SFR-supporting and SFR-non-interfering behavior of the SFR-enforcing subsystems.
ADV_TDS.2.6c The design shall summarize the behavior of the SFR-supporting subsystems.
ADV_TDS.2.7c The design shall provide a description of the interactions among all subsystems of the TSF.
ADV_TDS.2.8c The mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it.
ADV_TDS.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.2.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d The developer shall provide operational user guidance.
AGD_OPE.1.1c The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2c The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4c The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_OPE.1.5c The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6c The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7c The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1c The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2c The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2e The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Authorisation controls (ALC_CMC.3)

ALC_CMC.3.1d The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.3.2d The developer shall provide the CM documentation.
ALC_CMC.3.1c The TOE shall be labeled with its unique reference.
ALC_CMC.3.2c The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.3.3c The CM system shall uniquely identify all configuration items.
ALC_CMC.3.4c The CM system shall provide measures such that only authorized changes are made to the configuration items.
ALC_CMC.3.5c The CM documentation shall include a CM plan.
ALC_CMC.3.6c The CM plan shall describe how the CM system is used for the development of the TOE.
ALC_CMC.3.7c The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ALC_CMC.3.8c The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.
ALC_CMC.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 Implementation representation CM coverage (ALC_CMS.3)

ALC_CMS.3.1d The developer shall provide a configuration list for the TOE.
ALC_CMS.3.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation representation.
ALC_CMS.3.2c The configuration list shall uniquely identify the configuration items.

ALC_CMS.3.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.3 Delivery procedures (ALC_DEL.1)

ALC_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2d The developer shall use the delivery procedures.
ALC_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.4 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.
ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.2.3.5 Systematic flaw remediation (ALC_FLR.3)

ALC_FLR.3.1d The developer shall document flaw remediation procedures addressed to TOE developers.
ALC_FLR.3.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.3.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.3.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.3.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.3.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.3.5c The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.
ALC_FLR.3.6c The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.
ALC_FLR.3.7c The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.
ALC_FLR.3.8c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.3.9c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.
ALC_FLR.3.10c The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.
ALC_FLR.3.11c The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.
ALC_FLR.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.6 Developer defined life-cycle model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.
ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.
ALC_LCD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Tests (ATE)

5.2.4.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.
ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.2.2c The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.
ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.2 Testing: basic design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.
ATE_DPT.1.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.
ATE_DPT.1.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.
ATE_DPT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.3 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.
ATE_FUN.1.2d The developer shall provide test documentation.
ATE_FUN.1.1c The test documentation shall consist of test plans, expected test results and actual test results.
ATE_FUN.1.2c The test plans shall identify the tests to be performed and describe the scenarios for performing each test.
ATE_FUN.1.3c The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.4c The actual test results shall be consistent with the expected test results.
ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.4 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.
ATE_IND.2.1c The TOE shall be suitable for testing.
ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3e The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability analysis (AVA_VAN.2)

AVA_VAN.2.1d The developer shall provide the TOE for testing.

AVA_VAN.2.1c The TOE shall be suitable for testing.

AVA_VAN.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security audit

The TOE generates audit records for the following auditable events:

- Start-up and shutdown of the DBMS
- Use of special permissions (e.g. those often used by authorized administrators to circumvent access control policies)
- All auditable events for the 'minimum' level of audit as specified in the table below:

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1: Audit data generation	None	None
FAU_GEN_EXT.2: User and/or group identity association	None	None
FAU_SEL.1: Selective audit	All modifications to the audit configuration that occur while the audit collection functions are operating.	The identity of the authorized administrator that made the change to the audit configuration.
FDP_ACC.1: Subset access control	None	None
FDP_ACF.1: Security attribute based access control	Successful requests to perform an operation on an object covered by the SFP.	The identity of the subject performing the operation.
FDP_RIP.1: Subset residual information protection	None	None
FIA_ATD.1: User attribute definition	None	None
FIA_UAU.2: User authentication before any action	Unsuccessful use of the authentication mechanism	None
FIA_UID.2: User identification before any action	Unsuccessful use of the user identification mechanism including the user identity provided	None
FMT_MOF.1: Management of security functions behavior	None	None
FMT_MSA.1: Management of security attributes	None	None
FMT_MSA_EXT.3: Static attribute initialization	None	None
FMT_MTD.1: Management of TSF data	None	None
FMT_REV.1a: Revocation	Unsuccessful revocation of	Identity of individual attempting to

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
	security attributes.	revoke security attributes.
FMT_REV.1b: Revocation	Unsuccessful revocation of security attributes.	Identity of individual attempting to revoke security attributes.
FMT_SMF.1: Specification of Management Functions	Use of the management functions.	Identity of the administrator performing these functions.
FMT_SMR.1: Security roles	Modifications to the group of users that are part of a role.	Identity of authorized administrator modifying the role definition.
FPT_TRC_EXT.1: Internal TSF consistency	Restoring consistency.	
FTA_MCS.1: Basic limitation on multiple concurrent sessions	Rejection of a new session based on the limitation of multiple concurrent sessions.	
FTA_TAH_EXT.1: TOE access history	None	None
FTA_TSE.1: TOE session establishment	Denial of a session establishment due to the session establishment mechanism.	Identity of the individual attempting to establish the session.

Each audit record will include the date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event. In some cases, auditing can be configured to audit successful, unsuccessful, or both types of events; however, some events specifically audit either the success or failure of the event. The IT environment (specifically, the Operating System) provides protection, storage and the ability to view the audit records. It also provides the system clock information that is used by the TOE to timestamp each audit record. The audit records are stored on the local file system of the host. In a multi-host distributed architecture, where the Server subsystem of the TOE is run on a number of hosts, the audit records are stored on the local file system of the host on which the related auditable event is detected. Consequently, the aggregate audit record for an entire cluster may be distributed across multiple hosts, rather than being stored in a single location.

The TOE provides the Admin Interface, a web based browser GUI, through which an authorized administrator has the ability to configure the audit function to include or exclude auditable events based on group identity, event type, object identity and success or failure of the auditable security event.

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

- FAU_GEN.1: Audit records are generated for the appropriate security relevant events and include the date and time of the event, type of event, subject identity (if applicable) and outcome of the event.
- FAU_GEN_EXT.2: For applicable events, the TOE associates each auditable event resulting from actions of identified users with the identity of the user that caused the event.
- FAU_SEL.1: The TOE allows administrators to include or exclude auditable events based on group identity, event type, object identity and success and failure of auditable security events.

6.1.2 User data protection

The TOE enforces a Discretionary Access Control (DAC) Policy on all subjects, all Database Management System (DBMS) controlled objects and all operations among them. The DBMS controlled objects implemented by the TOE are *documents*. Documents are made up of one or more of the following:

- Content – XML, character, or binary content stored in a TOE database.
- Properties – XML describing the document properties.

- Locks – System-maintained XML describing the document locks. A lock can be exclusive or shared and can prevent update or deletion of content by other users.
- Other metadata – For example, document permissions.

Documents can be organized into *collections*, which are groups of related documents that enable queries to target subsets of content within the TOE. A document may belong to any number of collections simultaneously. A collection is implicitly created and exists in the system when a document in the system states that it is part of that collection.[6] Collections are not related to directories. They do not require member documents to conform to any URI patterns, they are not hierarchical and they cannot have properties set on them. The URIs that are used to name collections serve only as identifiers to the server. Collections do not have any security attributes associated to them; therefore the access control policy for them is the access control policy for the individual documents that are part of the collection. Access to each of the individual documents that belong to the specified collection is governed by that individual document's permissions.

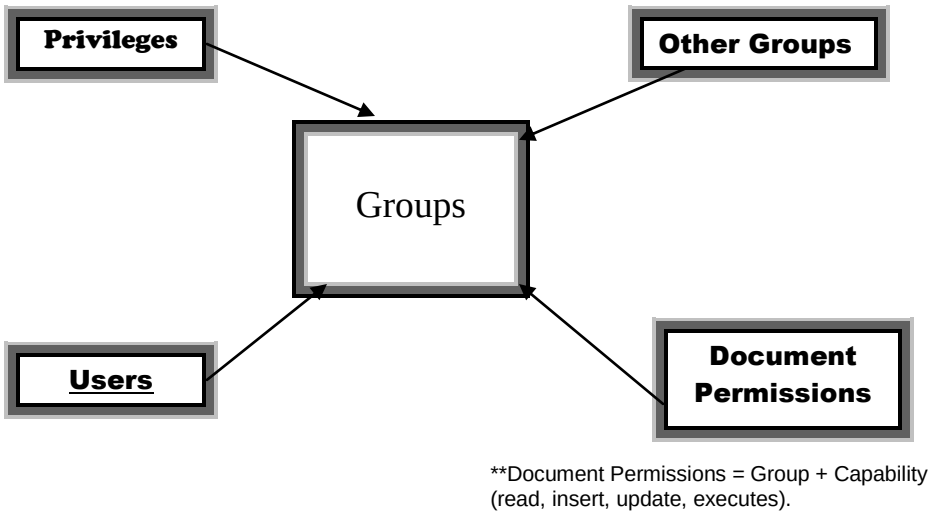
The DAC policy restricts access operations implemented for documents based on the document's identity (its Uniform Resource Identifier (URI)) and the user's authorized group membership. Users of the TOE are identified and authenticated by the TOE before any access to the system is granted. Once access to the system is granted, authorization provides the mechanism to control what functions a user is allowed to perform based on the user's group membership.

Access to all documents is denied unless access, based on group membership, is explicitly allowed. Documents are assigned permissions which are a combination of a group and a capability. Each permission associates a group with one of the following capabilities: Read, Update, Insert and Execute. Users assigned the group corresponding to the permission have the ability to perform the capability.

There are two types of privileges: Uniform Resource Identifier (URI) privileges and Execute privileges. URI privileges are used to control the creation of documents with certain URIs. Execute privileges are used to protect the execution of functions in XQuery code and to protect access to specific application servers.

Groups are the central point of authorization in the TOE. As shown in the diagram below, privileges (both execute and URI), document permissions, users and groups are all assigned to zero or more groups.

[6] An associated collection object is not created and stored in the security database unless it is protected. A collection created through the Admin Interface is a *protected collection* which is explicitly created and is stored in the security database. Protected Collections are optional and are not part of the evaluated configuration of the TOE.



By default, the DAC policy provides that documents are protected from unauthorized access according to the following ordered rules:

- If the requested mode of access is denied to every group of which the authorized user is a member, deny access;[7]
- If the requested mode of access is permitted to any group of which the authorized user is a member, grant access;
- Else, deny access.

Authorized administrators are able to bypass the DAC policy and therefore have explicitly authorized access to documents. Additionally, the TOE provides amplifications (referred to as *amps*) which allow users to assume additional privileges and permissions through temporary assumption of additional user groups during the execution of specified XQuery library functions. Amps can therefore be used to temporarily grant the authorized administrator role to an unprivileged user, thereby allowing them to bypass the DAC policy while performing specific functions. The effect of any additional permissions and privileges is limited to the specific function. Amps can only be configured and assigned by authorized administrators via access to the Admin Interface. Additionally, amplified functions are only located in either a designated administrator-controlled location in a directory on the MarkLogic Server Subsystem or in the database where they would be subject to the DAC policy and no user would have the ability to update or modify the function.

The TOE also ensures that any previous information content of a resource is made unavailable upon the allocation of the resource to TOE objects. Memory or disk space is only allocated when the size of the new data is first known, so that all previous data is overwritten by the new data.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.1: The TOE will enforce the DAC policy on all subjects, all documents and all operations among them.

[7] In the MarkLogic Server TOE, denial is implicit, therefore, the 'if' clause in this statement will never be true. According to the DBMS PP application note for FDP_ACF.1, the deny mode of access may be implicit.

- FDP_ACF.1: The TOE will enforce the DAC policy on documents based on the authorized user's group membership, the object identity and the access operations implemented for the documents. Documents will be protected from unauthorized access according to a set of ordered rules and only authorized administrators or users with temporarily elevated privileges shall be able to bypass the DAC policy.
- FDP_RIP.1: The TOE will ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to document objects.

6.1.3 Identification and authentication

The TOE maintains user accounts for the authorized users of the system and a list of security attributes for each user which includes the user's id, group membership, password and security-relevant database role. The TOE maintains the security relevant database role of authorized administrator. Authorized administrators are the only users that have permissions to manage the TOE security functions as described in this Security Target.

The TOE requires users to provide unique identification and authentication data (i.e. passwords) before any access to the system is granted. The TOE uses the digest authentication scheme, a commonly used web application authentication protocol, to provide encryption for passwords which are sent across the network as an MD5 hash using this scheme.[8] Digest authentication uses the browser's username and password prompt to obtain user credentials. The MarkLogic Server subsystem then authenticates the user credentials against the security database. Authentication simply verifies user credentials, associates that session with the authenticated user and determines their group membership. It does not grant any access or authority to perform any actions on the system. When a user logs into the TOE, their user id and password are validated against the security database.

All security attributes are stored in the security database of the MarkLogic Server subsystem. A single security database is associated with each HTTP or XDBC server. Where the TOE is configured with multiple servers, the same security database can be associated with the server or servers regardless of the number. The security database is accessed to authenticate users and to control user actions against the server.

Once access to the system is granted, authorization to access functions and data is implemented via the user's group membership. User groups are the central point of authorization in the TOE's security model. User groups are created with a specific set of privileges and permissions which apply to all users assigned to the group.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_ATD.1: The TOE maintains a list of security attributes for individual users.
- FIA_UAU.2: The TOE requires all users to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UID.2: The TOE requires all users to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.4 Security management

The TOE defines the security role of authorized administrator. Only authorized administrators can perform TOE security management functions. The Mark Logic TOE implements and enforces other user roles which translate into 'groups' for purposes of this ST and consistency with the DBMS PP terminology.

[8] For further information on digest authentication, please refer to RFC 2617. All Application Servers in the evaluated configuration must use digest based authentication.

The Admin Interface provides the interface through which the authorized administrator manages the security functions of the TOE. The Admin Interface provides administrator access to the following TOE security management functions:

- Management of User Accounts
  - Create, view, delete, and modify user accounts, including revoking security attributes associated with users.
  - Create, view, delete and modify privileges.
  - Create, view, delete and modify user groups.
- Management of Audit events
  - Enable and disable the audit configuration function.
  - Configure the audit function to include or exclude auditable events.
- Management of Access Control
  - Create, view and delete amps.
  - Create, query, modify or delete all the user and DBMS-controlled object security attributes associated with the DAC policy.
- Management of TOE sessions
  - Configure the limit on maximum number of concurrent sessions belonging to the individual user.
  - Configure the rules for denying session establishment.

The TOE provides administrators with the ability to revoke security attributes associated with users and objects. User security attributes are group membership, password and security relevant database role (i.e. authorized administrator role). DBMS-controlled object (i.e. document) security attributes are the access operations, or permissions that are implemented for the document.

Revocation of both object and user security attributes is enforced at all TOE interfaces based on the following rules:

- On the *revocation host*, revocation is effective on the next session that starts after the revocation request is committed.
- On other hosts in a cluster, revocation is effective no later than the receipt of the next *heartbeat* received from the *revocation host*.

The revocation hosts for object and user security attributes are different. The revocation host associated with revocation of user security attributes is the host on which the security forest resides. The revocation host associated with revocation of document security attributes is the host on which the document resides locally. A heartbeat is a cluster synchronization message and occurs once per second.

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

- FMT_MOF.1: The TOE restricts the ability to disable and enable the audit configuration function to authorized administrators.
- FMT_MSA.1: The TOE enforces the DAC policy to restrict the ability to manage the security attributes to authorized administrators.
- FMT_MSA_EXT.3: The TOE enforces the DAC policy to provide restrictive default values for security attributes.
- FMT_MTD.1: The TOE restricts the ability to include and exclude auditable events to authorized administrators.

- FMT_REV.1a: Only TOE administrators can revoke user security attributes according to enforceable rules.
- FMT_REV.1b: Only TOE administrators can revoke object security attributes according to enforceable rules.
- FMT_SMF.1: The TOE performs the following security management functions: management of audit events, management of user accounts, management of access control, management of TOE sessions.
- FMT_SMR.1: The TOE maintains the security role of authorized administrator.

6.1.5 Protection of the TSF[9]

The TOE provides security mechanisms for its security functions to ensure that it can protect itself from tampering and bypass by untrusted entities. One of the protection mechanisms is that users must authenticate before any administrative operations can be performed on the system. The TSF requires that all users be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user. The TOE also enforces an access control policy which restricts user access to DBMS-controlled objects. Authorized users only have access to functions as specified by their assigned group membership and capabilities. The Operating System in the IT environment of the TOE provides an execution environment that ensures thread separation and safeguards the results of one query from interfering with the results of another query. The TOE also relies on the IT environment for process isolation.

The TOE also has the ability to replicate data by propagating updated configuration and security files throughout a cluster. Configuration information includes the Cluster, Host, Cluster Management Group, Forest, and Database information as described in Section 2.2 above. The TOE ensures the consistency of TSF data between parts of the TOE for both configuration information and security information as follows:

- The TOE's configuration information is stored in a set of files in a special file system directory structure on each host in a cluster. For example, on Linux, the default location for configuration files is /var/opt/MarkLogic. Each configuration file contains a configuration file system timestamp which is a monotonically-increasing number that increases with every configuration or content change cluster-wide. The *configuration file system timestamp* is the latest timestamp at the time the file was last updated. Each heartbeat, or cluster synchronization message, that occurs once per second, contains the *heartbeat configuration system timestamp* which is the most recent timestamp of the configuration files of the host from which it was issued. Within one second of receipt of a heartbeat, the receiving host examines the heartbeat configuration system timestamp and if it is more recent than its own, the newer configuration files from the host that issued the heartbeat are copied and the local configuration files are replaced with the newer versions.

- The TOE's security data is stored in the security database. There is one security database per cluster and other hosts in the cluster cache some of the documents in this database to the security database cache. There is a timestamp for both the security database and the security database cache which indicates the time of the most recent change to the database or database cache. Each heartbeat also contains a copy of the security timestamp. Upon receipt of a heartbeat, if that heartbeat contains a security timestamp more recent than the security timestamp on the receiving host's security database cache, then the receiving host's database cache is invalidated. Consequently, all sessions initiated on that host subsequent to the security database cache flush will be forced to retrieve the latest copies of documents from the security database.

[9] The CC v3.1 requires that the TOE summary specification includes a description of how the TOE protects itself from interfering, logical tampering and bypass (CEM, ASE_TSS.2.2C and 2.3C). Further information regarding domain separation, logical tampering and bypass protection can be found in the TOE Design and Security Architecture documents.

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

- FPT_TRC_EXT.1: The TOE ensures that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.

6.1.6 TOE access

The TOE restricts the maximum number of concurrent sessions that belong to the same user. This is enforced by the setting of an administrator configurable number of sessions per user. Upon successful session establishment, the TOE will store and retrieve the date and time of the last successful session establishment, the date and time of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment to the user. TOE session establishment history and data is stored in the last-login database of the TOE and is persisted indefinitely. Session establishment data is maintained on a per user basis across the entire cluster. Therefore, within a given cluster, session establishment data for any hosts that have been previously accessed by a user will be reported to the user from any other hosts subsequently accessed within the same cluster.

The TOE provides session establishment control and can deny session establishment based on either user identity or group membership, or time of the day or day of the week or some combination thereof. Authorized administrators can configure the session establishment rules via the Admin Interface. Rules for session denial are configured for each application server (i.e. HTTP or XDBC). Session establishment may also be denied if the user does not have the execute privilege required to establish a session on the application server to which the user is attempting to connect. Execute privileges can be optionally used to control access to an HTTP or XDBC server. The Admin Interface provides the option of specifying a privilege required for server access. If such a privilege is specified, any users that access the server must possess the specified privilege.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_MCS.1: The TOE will restrict the maximum number of concurrent sessions that belong to a user by enforcing an administrator configurable limit on the number of sessions per user.
- FTA_TAH_EXP.1: Upon successful session establishment, the TOE shall store and retrieve to the user, the date and time of the last successful session establishment, the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.
- FTA_TSE.1: The TOE denies session establishment based on user identity or group membership, or time of day, or day of week or by application server privilege or a combination thereof.

7. Protection Profile Claims

This section provides the PP conformance claims.

7.1 PP Identification

The TOE conforms to the U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2, July 25, 2007.

7.2 PP Tailoring and Conformance Rationale

The DBMS PP is written in conformance with the CC v3.1, Parts 2 and 3. This ST claims conformance to both the DBMS PP and the CC v3.1, Revision 2.

In CC Part 1, CC v3.1, it states that an ST is equivalent or more restrictive than a PP if:

- Paragraph 444 (Per PD-0137): "All TOEs that meet the ST, also meet the PP."
- Paragraph 444 (Per PD-0137): "All operational environments that would meet the security problem definition in the ST, would also meet the security problem definition in the PP."
- Paragraph 445 (Per PD-0137): "All operational environments that would meet the security objectives for the operational environment in the ST would also meet the security objectives for the operational environment in the PP. "

Below is a bulleted summary of all of the sections in the ST that include modifications to the PP. Each bulleted item includes the section number where the full rationale can be found.

The Security Environment, Objectives, and Requirements in this ST have been reproduced from the DBMS PP as indicated below:

- The assumptions, threats and organizational security policies are reproduced directly from the DBMS PP. With the exception of two new threats, T.OPS and T.PRIV, and three new assumptions, A.ADMIN, A.AUTH and A.BROWSER there are no modifications to the security environment of the PP. (**Section 3**)

- The security objectives are reproduced directly from the DBMS PP. With the exception of four new objectives, O.ACCESS, O.PROTECT, OE.AUTH and OE.BROWSER, there are no modifications to the security objectives of the PP. (**Section 4**)

- There are no modifications to the DBMS PP security objectives, assumptions, threats or organizational policies; therefore, the rationale in the DBMS PP is valid for this ST. There is, however, additional rationale provided for the mapping of the three assumption, two threats and four objectives added to this ST. (**Section 8.1**)

- Apart from reference numbers, headings and minor wording changes to the Security Functional requirements to bring them in compliance with CC v3.1, there is only one modification to the TOE Security Functional Requirements from the DBMS PP.

    - The FIA_UAU.2 and FIA_UID.2 requirements have been added to the security functional requirements in Section 5 to address the fact that the TOE itself implements the authentication mechanism. (**Section 8.2.1**)

- There is one modification to the IT Environment Security Functional Requirements from the DBMS PP. The extended requirement FIT_PPC_EXP.1 has been removed. In CC v3.1, there is no distinction between IT and non-IT environment and there are no IT environment SFRs. The responsibility for the IT environment is assigned to objectives and assumptions. (**Section 8.2.1**)

- The TOE Security Assurance Requirements in the DBMS PP have been replaced and augmented in this ST with the CC v3.1, Part 3, EAL requirements augmented with ALC_FLR.3. (**Section 8.2.2**)

- The Security functional and Assurance requirements rationale table from Section 6.3 of the DBMS PP is applicable to the requirements in this ST and is therefore valid for this ST. (**Section 8.2**).

- The requirements dependency table demonstrates the dependencies for the CC v3.1 EAL 3 assurance requirements which are equivalent and/or exceed the EAL 2 requirements in the DBMS PP. All of the dependencies among the claimed security requirements are satisfied for EAL 3 and therefore the requirements work together to accomplish the overall objectives defined for the TOE. (**Section 8.3**)

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;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

The security objective rationale is presented in Section 6.1 and Section 6.2 of the DBMS PP. All of the assumptions, threats, organizational policies and security objectives have been reproduced from the DBMS PP to this ST. Therefore, the rationale in the DBMS PP is valid for this ST.

The following mapping is provided as additional rationale for the assumptions (A.ADMIN, A.AUTH and A.BROWSER), the threats (T.OPS and T.PRIV) and objectives (O.ACCESS, O.PROTECT, OE.AUTH and OE.BROWSER) which have been added to the ST. Refer to Section 8.2.1 below for the mapping of the TOE objectives to SFRs.

8.1.1.1 A.ADMIN

The Admin Interface application runs on Port 8001 behind a firewall which is configured to block egress and ingress of traffic over Port 8001.

This Assumption is satisfied by ensuring that:

- OE.NOEVIL: Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance including guidance on configuring the Admin Interface for the evaluated configuration.
- OE.PHYSICAL: Physical security will be provided within the domain for IT assets and stored, processed, and transmitted information. This security will include a firewall configured to block egress and ingress of traffic over Port 8001, behind which the Admin Interface shall run.

8.1.1.2 A.AUTH

Passwords are encrypted during the authentication process.

This Assumption is satisfied by ensuring that:

- OE.AUTH: Password encryption during the authentication process is provided by the IT environment of the TOE, the Internet Explorer web browser.

8.1.1.3 A.BROWSER

The web browsers used to access the Admin Interface perform correctly such that when the browser is closed, the active Admin session is terminated.

This Assumption is satisfied by ensuring that:

- OE.NOEVIL: Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance including guidance on configuring and using the Admin Interface for the evaluated configuration.
- OE.BROWSER: The web browsers used to access the Admin Interface will perform correctly and when the Administrator closes the browser, the active Admin session will be terminated such that when the browser is restarted, the user will be prompted to authenticate with a username and password.

8.1.1.4 T.OPS

An unauthorized process or application may gain access to the TOE security functions and data, inappropriately changing the configuration data for the TOE security functions.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE must only allow authorized users and processes to access protected TOE functions and data.
- O.PROTECT: The TOE must protect its functions and data from unauthorized modifications and access.

8.1.1.5 T.PRIV

An unauthorized user may gain access to the TOE and exploit system privileges to gain access to TOE security functions and data, inappropriately performing changing the configuration data for TOE security functions.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE must allow only authorized users and processes to access protected TOE functions and data.
- O.PROTECT: The TOE must protect its functions and data from unauthorized modifications and access.

8.2 Security Functional and Assurance Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. The security requirements rationale is found in Section 6.3 of the DBMS PP. With the exception of those items noted below in section 8.2.1, all of the rationale found in the DBMS PP is valid for the security functional requirements in this ST.

8.2.1 Security Functional Requirements Rationale

Apart from reference numbers, headings and minor wording changes to the Security Functional requirements to bring them in compliance with CC v3.1, the following modifications have been made to the IT Security Requirements drawn from the DBMS PP in order for this ST to conform to CC v3.1:

- The FIT_PPC_EXP.1 (IT Environment Protection Profile Compliance) extended IT Environment Security Functional Requirement has been removed. This requirement is defined as follows:

  > **FIT_PPC_EXP.1.1** The IT environment shall be compliant with the requirements of the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater.

  For CC v3.1, there is no distinction between IT and non-IT environment and there are no IT environment SFRs. The responsibility for the IT environment is assigned to objectives and assumptions. This is explained in Part 1 of the CC v3.1 which states that SFRs are a translation of

the security objectives of the TOE. There is no translation required for security objectives of the operational environment because the operational environment is not evaluated and therefore does not require a description aimed at its evaluation. It may be the case that parts of an operational environment are evaluated in another evaluation, but this is out of the scope for the current evaluation.

Therefore, in CC v3.1, the DBMS PP environment objective, OE.OS_PP_VALIDATED (The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness) which maps to the assumption, A.OS_PP_VALIDATED (The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness) requires no translation to an IT security functional requirement and is adequate and equivalent in meaning and scope to represent the extended requirement (FIT_PPC_EXP.1) as found in the DBMS PP.

- FIA_UID.2 and FIA_UAU.2 have been added to the security functional requirements in this ST. According to CCEVS Policy Letter 13 and its addendum, security functionality to be included in the TOE is to be whatever is advertised to potential customers in addition to the security functionality that customers would expect based on product type. Authentication is provided by the TOE and therefore should have requirements which can be tested.

The FIA_UAU.2 and FIA_UID.2 SFRs map as follows to the O.ACCESS and O.PROTECT objectives:

8.2.1.1 O.ACCESS

The TOE must allow only authorized users and processes (applications) to access protected TOE functions and data.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UAU.2: The TOE requires each user to be successfully authenticated before allowing any TSF mediated actions on behalf of that user.
- FIA_UID.2: The TOE shall require each user to be successfully identified before allowing any TSF mediated actions on behalf of that user.

8.2.1.2 O.PROTECT

The TOE must protect its functions and data from unauthorized access and modifications.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UAU.2: The TOE requires all users to be successfully authenticated before allowing any other TSF mediated actions on behalf of the user.
- FIA_UID.2: The TOE requires all users to be successfully identified before allowing any other TSF mediated actions on behalf of the user.

8.2.2 Security Assurance Requirements Rationale

The assurance requirements in the DBMS PP are the CC v3.1 Part 3 requirements for EAL 2 augmented with ALC_FLR.2. The assurance requirements of this ST are the CC v3.1, Release 2, Part 3 EAL 3 requirements augmented with ALC_FLR.3. The assurance requirements in this ST are conformant to the DBMS PP, in that they meet or exceed those requirements.

Mark Logic has elected to pursue a more rigorous assurance level, increased from EAL2 as specified in the DBMS PP to EAL3, as specified in section 1.2 of this ST. EAL3 was selected as the assurance level because the TOE is a commercial product whose users require a moderate to high level of independently assured security. The TOE is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have little attack potential. As such, EAL3 is appropriate to provide the assurance necessary to counter the limited potential for attack.

While the EAL chosen is not the same as is specified in the DBMS PP, this ST remains DBMS PP conformant because the EAL chosen in this ST (EAL 3) is hierarchical to the EAL specified in the DBMS PP. ALC_FLR.3 is also hierarchical to ALC_FLR.2.

In accordance with CCEVS Policy Letter #3, it is acceptable for a Security Target to contain a superset of a PP's requirements and still be in compliance with the PP. There are several ways in which an ST can be a superset of a PP as follows:

- It can include functional components that are hierarchical to those of the PP.
- It can include functional requirements in addition to those specified in the PP.
- It can specify a higher EAL level than that required for compliance with the PP.
- It can specify assurance components in addition to those required for the PP.

The additional capabilities introduced by the higher assurance requirements and the additional functional requirements in this ST do not introduce any security vulnerabilities nor circumvent or interfere with required security functions. Therefore, this ST remains compliant with the DBMS PP.

8.3 Requirement Dependency Rationale

The modifications to the security functional requirements that have been noted in previous sections do not introduce any additional dependencies. The dependencies for the CC v3.1 SFRs are the same as those reproduced from the DBMS PP.. The rationale for satisfying all dependencies is presented in Section 6.5 of the DBMS PP.

Based on rationale provided in previous sections of this Security Target **(Section 8)**, the CC v3.1 EAL3 SARs are equivalent to and/or exceed the requirements of those from the DBMS PP. This table demonstrates the dependencies for the CC v3.1 requirements and shows that all dependencies among the claimed security requirements are satisfied for EAL3 and therefore the requirements work together to accomplish the overall objectives defined for the TOE.

The following table identifies each security functional and assurance requirement in this ST. The table enumerates the dependencies of each requirement as specified in the CC and then identifies the requirement in this ST that satisfies each of those dependencies. Note that in some cases a dependency is satisfied by a hierarchically (as defined in the CC) greater requirement component (identified in bold). The ST Dependencies indicated in italics are dependencies satisfied by the IT environment of the TOE.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	*FPT_STM.1*
FAU_GEN_EXT.2	FAU_GEN.1 and FIA_UID.2	FAU_GEN.1 and **FIA_UID.2**
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1 and FMT_MTD.1
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1 and the dependency on FMT_MSA.3 is satisfied by FMT_MSA_EXT.3
FDP_RIP.1	none	none
FIA_ATD.1	none	none
FIA_UAU.2	FIA_UID.1	**FIA_UID.2**
FIA_UID.2	None	none
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1	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.1
FMT_MSA_EXT.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_REV.1a	FMT_SMR.1	FMT_SMR.1
FMT_REV.1b	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	none	none

ST Requirement	CC Dependencies	ST Dependencies
FMT_SMR.1	FIA_UID.1	**FIA_UID.2**
FPT_TRC_EXT.1	FPT_ITT.1	*FPT_ITT.1*
FTA_MCS.1	FIA_UID.1	**FIA_UID.2**
FTA_TAH_EXT.1	none	none
FTA_TSE.1	none	none
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	**ADV_FSP.3** and **ADV_TDS.2**
ADV_FSP.3	ADV_TDS.1	**ADV_TDS.2**
ADV_TDS.2	ADV_FSP.3	ADV_FSP.3
AGD_OPE.1	ADV_FSP.1	**ADV_FSP.3**
AGD_PRE.1	none	none
ALC_CMC.3	ALC_CMS.1 and ALC_DVS.1	**ALC_CMS.3** and ALC_DVS.1
ALC_CMS.3	none	none
ALC_DEL.1	none	none
ALC_DVS.1	none	none
ALC_FLR.3	none	none
ALC_LCD.1	none	none
ATE_COV.2	ADV_FSP.2 and ATE_FUN.1	**ADV_FSP.3** and ATE_FUN.1
ATE_DPT.1	ADV_ARC.1 and ADV_TDS.2 and ATE_FUN.1	ADV_ARC.1 and ADV_TDS.2 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	**ATE_COV.2**
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	**ADV_FSP.3** and AGD_OPE.1 and AGD_PRE.1 and **ATE_COV.2** and ATE_FUN.1
AVA_VAN.2	ADV_ARC.1 and ADV_FSP.1 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and **ADV_FSP.3** and **ADV_TDS.2** and AGD_OPE.1 and AGD_PRE.1

8.4 Extended Requirements Rationale

There are no extended requirements beyond those in the PP. The extended requirements rationale is presented in Section 6.6 of the DBMS PP.

8.5 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 3 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	User data protection	Identification and authentication	Security management	Protection of the TSF	TOE access
FAU_GEN.1	X					
FAU_GEN_EXT.2	X					
FAU_SEL.1	X					
FDP_ACC.1		X				
FDP_ACF.1		X				
FDP_RIP.1		X				
FIA_ATD.1			X			
FIA_UAU.2			X			
FIA_UID.2			X			
FMT_MOF.1				X		
FMT_MSA.1				X		
FMT_MSA_EXT.3				X		
FMT_MTD.1				X		
FMT_REV.1a				X		
FMT_REV.1b				X		
FMT_SMF.1				X		
FMT_SMR.1				X		
FPT_TRC_EXT.1					X	
FTA_MCS.1						X
FTA_TAH_EXT.1						X
FTA_TSE.1						X

Table 3 Security Functions vs. Requirements Mapping

8.6 PP Claims Rationale

See Section 7, Protection Profile Claims.

9. Appendix

9.1 Organizational Policies Not Applicable to the TOE

In accordance with the DBMS PP which follows the instructions of the Consistency Instruction Manual for Development of U.S. Government Protection Profiles for use in Basic Robustness Environments, Version 3.0 (CIM), the following organizational policies have been included here.

9.1.1 Organizational Policies Not Applicable to the TOE

P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. This threat is not applicable to the TOE due to the absence of a client interface that is capable of displaying an access banner.
P.CRYPTOGRAPHY	Only NIST FIPS validate cryptography (methods and implementations) are acceptable for key management. This threat is not applicable to the TOE due to the absence of cryptographic requirements for the TOE.