IBM DB2 Version 9.7 Enterprise Server Edition for Linux, Unix, and Windows Security Target

Revision 1.0
August 3, 2009

Prepared for:

IBM Canada, Ltd.
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
Canada

Prepared By:

An Employee-Owned Company

Science Applications International Corporation

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Security Target Introduction

This Security Target (ST) describes the IT security requirements for the IBM DB2 Enterprise Server Edition Version 9.7 for Linux, Unix, and Windows; herein collectively referred to as DB2. DB2 is a Relational Database Management System (RDBMS) developed by IBM Canada, Ltd., 3600 Steeles Avenue East, Markham, Ontario L3R 9Z7, Canada and sold by IBM Corporation, Route 100, Somers, NY, USA 10589.

DB2 has historically been developed with a goal of fulfilling the C2 requirements of the Trusted Computer System Evaluation Criteria (TCSEC; also known as the "Orange Book"). The Common Criteria (CC) for Information Technology Security Evaluation eventually replaced the TCSEC and the C2 TCSEC requirements have been recast in the Controlled Access Protection Profile (CAPP). As a result, the security environment, security objectives, and security requirements are derived largely from the CAPP. However, since DB2 is a RDBMS and not a complete Operating System, some of the requirements have been assigned to the operational environment (i.e., underlying operating system) and conformance cannot, therefore, be claimed in this ST. At this point, a number of subsequent versions of DB2 have been successfully evaluated using the CC.

Note that while there are DBMS-specific PPs, conformance cannot be achieved due to certain requirements that could be met by few if any current DBMS products. Or particular note are the requirements dictating that the DBMS must be able to limit the number of concurrent connections a given user can have; the DBMS must store and retrieve date/time information associated with sessions, and the DBMS must be able to restrict sessions based user/group identity, time of day and day of week.

1.1 Security Target, TOE and CC Identification

ST Title – IBM DB2 Version 9.7 Enterprise Server Edition for Linux, Unix, and Windows Security Target

ST Version – Revision 1.0

ST Date – August 3, 2009

TOE Identification – IBM DB2 Enterprise Server Edition Version 9.7 for Linux, Unix, and Windows

- The TOE can optionally be configured with or without the IBM Base Warehouse Feature for DB2 9.7 license option.
- The TOE can optionally be configured to trust other servers using Trusted Contexts.
- The TOE can optionally be configured to utilize SSL when connecting with Java-based clients.
- The TOE can alternately be configured to utilize authentication services of its underlying operating system, an external LDAP server, or an external KDC server.
- While the TOE can be configured to use DES to protect authentication credentials, that mechanism has not been subject to evaluation and as such should not be solely relied upon as an adequate means of protection. Note that DES is considered a weak algorithm and is no longer FIPS approved.
- While the TOE can be installed using a non-root install option, that configuration does not include all the features available to DB2 and has not been subject to evaluation. As such, this evaluation applies only to a normal (root) installation of DB2.
- While users can employ user defined function (UDF) options to encrypt and decrypt data, this mechanism has not been subject to evaluation and as such should not be solely relied upon as an adequate means of protection.

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

1.2 Conformance Claims

This TOE conforms 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
- Package Conformance:
  - Evaluation Assurance Level 4 (EAL 4) augmented with ALC_FLR.1

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the ST.

1.3.1 Conventions

The following conventions have been applied in this document:

- All requirements in this ST are reproduced relative to the requirements defined in CC v3.1r2.
- 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 identifier. 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 in cases where a selection operation is combined with an assignment operation and the assignment is null, the assignment operation is simply deleted leaving only the completed selection to identify the combination of operations. Alternately, if the assignment is not null the assignment is identified with embedded brackets which are bolded and italicized (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 requirements (i.e., those not taken from the CC) are identified by '_EXP' appearing as an element of the requirement label. Note that the extended requirements are also identified in Section 8.5.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Acronyms

CC	Common Criteria
CM	Configuration Management
DAC	Discretionary Access Control
DDL	Data Definition Language
DML	Data Manipulation Language
DRDA	Distributed Relational Database Architecture
IBM	International Business Machines
LBAC	Label Based Access Control
OS	Operating System
PP	Protection Profile
RDBMS	Relational Database Management System
SQL	Structured Query Language
ST	Security Target
TCSEC	Trusted Computer System Evaluation Criteria
TSF	TOE Security Features
TSP	TOE Security Policy
TOE	Target Of Evaluation

1.4 Security Target Overview and Organization

The DB2 Target of Evaluation (TOE) is a Database System offering a wide range of database related services. This ST describes the DB2 TOE, intended environments, security objectives, security requirements (for the TOE), security functions, and all necessary rationale. This information is organized into the following additional sections:

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

2. TOE Description

DB2 is a relational database management system (RDBMS) provided by IBM. As a RDBMS, DB2 supports the Structured Query Language (SQL) interface from a client that is connected to the database server. From the client, commands can be entered interactively or through an executing program to the database server to create databases, database tables, and to store and retrieve information from tables. DB2 can be installed on a number of possible operating environments.

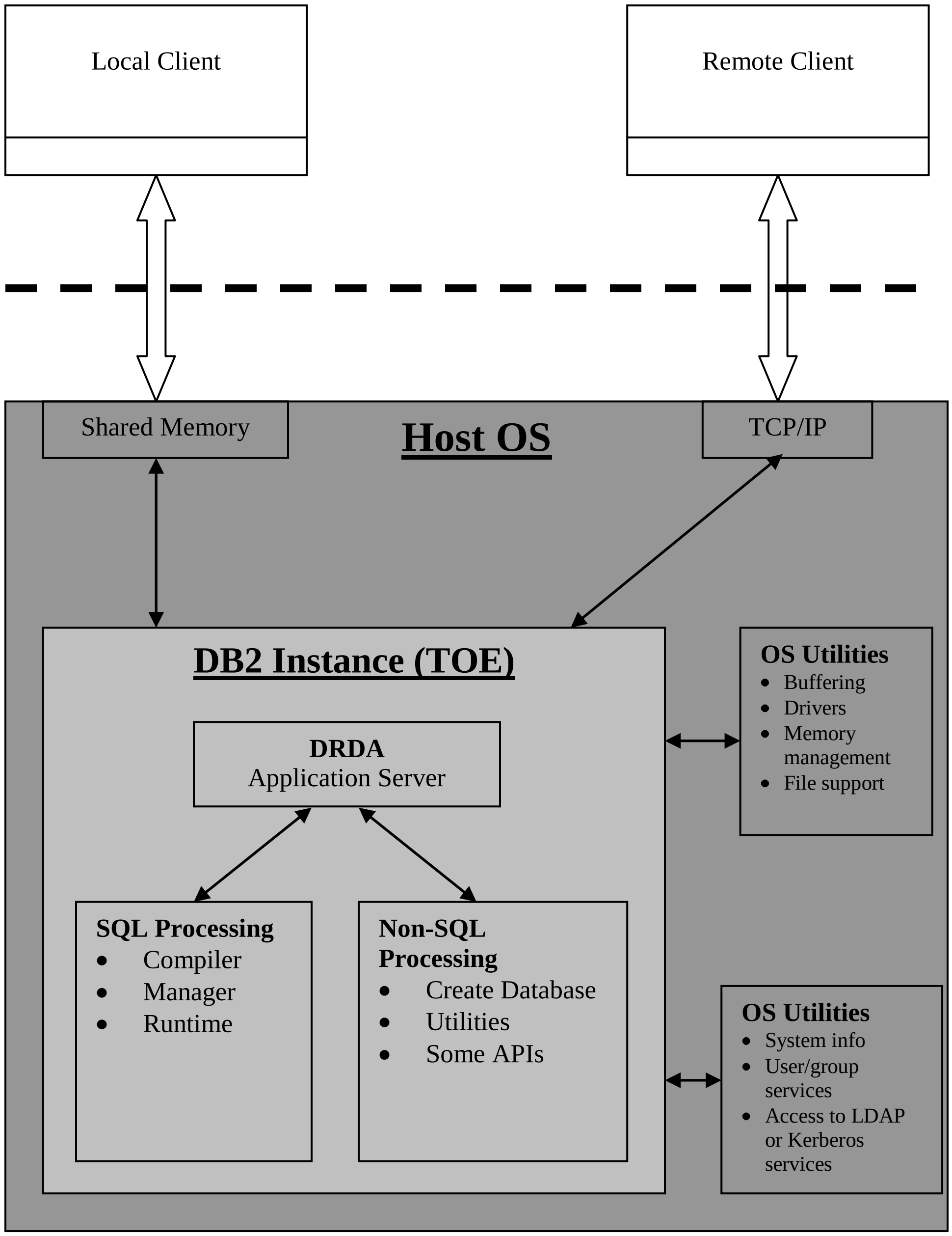

Figure 1 TOE Security Environment

The configuration in which the DB2 application is evaluated is described in Figure 1 TOE Security Environment.

The TOE for the DB2 configuration includes all components within the lightly shaded box entitled "DB2 Instance" And the TOE includes one or more such instances. For the purposes of this ST, the TOE is at least one partition, but can be distributed across a number of logically (i.e., on the same underlying machine) or physically (i.e., on different underlying machines) separate partitions. This latter feature is known as Database Partitioning Feature (DPF). From the user perspective, there is effectively no difference, while the distributed partitions work in concert to answer user queries. In relation to the figure above, a 'DB2 Instance' may actually be distributed across multiple DB2 partitions acting together to form that logical view.

The Host Operating System (OS), inside the darkly shaded box, is part of the TOE's operational environment (i.e., it is not part of the product or TOE). Similarly, an LDAP or KDC server (not pictured) could be configured to provide authentication services and those servers also would be in the TOE's operational environment. The DB2 software is tightly linked to the OS and in some cases, security functions are allocated to the OS (or other authentication servers), as appropriate. For the purposes of this ST, the OS is AIX 6, Linux RHEL 5 update 2, Linux SLES 10 with SP2, Microsoft Windows Server 2003 Enterprise Edition with SP2, or Solaris 10. Any standard-compliant LDAP or KDC server can also be used in conjunction with the TOE.

The clients, inside the unshaded boxes, are client applications either operating in the context of the host OS using shared memory to communicate with the TOE or in the context of some other OS using a network connection tom communicate with the TOE. While the transport may differ, the application layer communication is the same in either case.

2.1 Product Type

DB2 is a multi-user RDBMS that operates in the context of a hosting operating system and allows authorized users[1] to create and manage databases.

DB2 operates as a set of applications (e.g., servers) in an operational environment consisting of all software residing on the host platform(s) but not part of the DB2 TOE. For the purposes of this discussion it is referred to as the Host OS (refer to Figure 1 TOE Security Environment). The operational environment, including the Host OS and optionally an LDAP or KDC authentication server, provides fundamental supporting mechanisms to the TOE. In particular, it supplies a trusted authentication mechanism and utilities to manage system resources and I/O channels.

2.2 Product Description

This section describes the basic functions performed within DB2. These functions are depicted as individual blocks within the DB2 Instance (TOE) box in Figure 1 TOE Security Environment. DB2 is implemented using the concept of a DB2 instance where an instance is a complete environment dictating what can be done to data and managing the system resources assigned to it. A DB2 instance has one or more databases under its control that cannot be directly accessed by any other instance.

DB2 implements a Discretionary Access Control (DAC) security policy by default. This permits a confidential security mechanism to ensure data is protected against unauthorized or accidental disclosure or destruction. Auditing is supported at the DB2 instance level meaning that all modules within the TOE are capable of creating audit events. Review and analysis of the audit logs is restricted to users with appropriate authority or privilege; system administrator authority for instance level audit logs, security administrator for database-specific audit logs, or execute privilege to available audit functions granted by a security administrator.

2.2.1 DRDA Protocol Handler

The DRDA Application Server (AS) module within DB2 allows for DB2 to act as an Application Server within the Distributed Relational Database Architecture (DRDA). DRDA is an OpenGroup standard used in the management of distributed data. The DB2 DRDA AS module architecture provides support for one or more DRDA Application Requestors (DRDA AR), commonly referred to as clients, to access a specific DB2 instance or DB2 database and issue SQL and non-SQL requests against that object. Upon initiation of communication between a client and the

[1] The term *authorized user* is used to generally refer to a user authorized (e.g., by access permissions) to perform a corresponding function depending on the context in which the term is used.

DB2 DRDA AS module, a common "security mechanism" is negotiated. This mechanism may be one of a number of different security protocols; for the purpose of this TOE, the only allowed security mechanism is the "Userid, Password" mechanism as described in the DRDA standard. If validation of the password fails, the DRDA AS terminates conversation with the client that provided the failed password. If the password is authenticated, a DRDA session, or connection, is established and the client may begin to pass requests to DB2 for processing. These requests are of two general types: SQL requests, which are handled by the DB2 SQL Processing module, and non-SQL requests, which are handled by the DB2 Non-SQL Processing module. The DRDA AS module identifies the type of request and passes it to the appropriate module for further processing.

2.2.2 SQL Processing

The DB2 SQL Processing module is responsible for the analysis and execution of client requests related to the processing of Structured Query Language (SQL) statements. DB2 supports the ANSI/ISO SQL2 standard for all types of SQL statements including:

- Data Definition Language (DDL) statements that create, alter, drop, rename, or transfer ownership of database objects.
- Data Manipulation Language (DML) statements that are used to query or modify the data contained within database objects. Modification can occur in one of three ways: row insertion, row deletion, or row modification via column updates. These statements include SELECT, INSERT, UPDATE, and DELETE SQL statements.
- GRANT and REVOKE statements that are used to control the access to database authorities as well as privileges on database objects
- Transaction control statements that are use to manage the integrity of the database with respect to any modification made by a client. These statements include, among others, the ROLLBACK and COMMIT SQL statements..
- Miscellaneous statements used to perform a number of different actions on database objects or on the connection environment.

The DB2 SQL Processing module is comprised of three distinct components: the SQL Manager, the SQL Compiler, and the SQL Runtime components. The responsibilities of these components as they relate to the processing of SQL statements is described in the following sections.

2.2.2.1 SQL Manager

The SQL Manager is responsible for accepting SQL requests from the client, validating them, and then coordinating any subsequent processing of the request to ensure it is properly answered. The SQL Manager can accept SQL requests related to static or dynamic SQL statements. Static SQL statements have their contents made known to DB2 prior to the request arriving from the client through a process called "binding" which results in the statement being compiled by the SQL Compiler and the resultant information being stored in the DB2 system catalogs for later use. Dynamic SQL statements are unknown to DB2 until the request arrives at which time they are compiled by the SQL Compiler. The information produced by the SQL compiler contains the executable form of the statement, referred to as a section, a list of the required privileges for any client wishing to run the section as well as a list of the database objects upon which the section is dependent for its execution integrity.

The SQL Manager processes SQL requests from a client by matching the request to a specific SQL statement. Once the statement has been identified and its related information acquired, either from the DB2 system catalogs or the SQL Compiler, the SQL Manager then enforces the discretionary access control policy by ensuring that the required privileges for the section are held by the primary authorization name (a specific user identifier), or by any relevant secondary authorization names (the identifiers for any relevant groups to which the primary authorization name belongs and roles to which any other authorization name belongs[2]), associated with the request from the client. If the privileges are held, then the section is passed to the SQL Runtime component for execution.

[2] Note that users, groups, and other roles can be assigned to roles. As such, role hierarchies are supported.

2.2.2.2 SQL Compiler

The SQL Compiler is responsible for analyzing an SQL statement and producing an efficient executable form of that statement, called a "section", as well as additional information about that section such as its object dependencies and required privileges. The SQL Compiler parses an SQL statement into an internal representation, or model, of the statement that is then used to analyze the scope and intent of the statement. Additional information is added to the internal model, where appropriate, from the DB2 system catalogs in order to properly represent the full extent of the statement's use of any database objects. Once complete, the internal model is then analyzed and optimized in order to produce the most efficient plan to satisfy the statement. The SQL Compiler then generates an executable form of the statement using the internal DB2 constructs and operators used by the SQL Runtime component.

2.2.2.3 SQL Runtime

The SQL Runtime component is responsible for the actual execution of the section related to the request and the production of any response to the client required by the request. The success or failure of the actual execution as well as any additional response is given back to the SQL Manager for return to the client.

2.2.3 Non-SQL Processing

The DB2 Non-SQL Processing module is responsible for the analysis and execution of all those client requests not concerned with SQL statements. Such requests are used to invoke a number of Application Program Interfaces (APIs) and utilities provided by DB2 that do not use SQL statements to perform their specified actions. There exist a number of these APIs and utilities at both the DB2 Instance level as well as at the individual database level within a DB2 instance. Each API and utility provided by DB2 has an assigned privilege or authority requirement as defined by DB2. The DB2 Non-SQL Processing module enforces the discretionary access control policy for these non-SQL requests by ensuring that the required privilege or authority is held by either the primary authorization name, or secondary authorization names where applicable, of the requestor.

2.2.4 Optional Features

DB2 includes some additional security-relevant features that are enabled only when an applicable feature is configured. Since these features are not necessary available in every product configuration, they are considered optional.

2.2.4.1 Database Partitioning Feature (DPF)

DB2 includes an optional data partitioning feature (DPF). This feature allows DB2 to be instantiated across multiple partitions (on the same or separate machines) for the purpose of scalability (e.g., more computational and storage resources). The overall security mechanisms remain the same, though processing may be spread across the partitions internally. Note that this feature is included within the IBM Base Warehouse Feature for DB2 9.7 which can optionally be purchased as a license option in addition to the base DB2 product.

2.2.4.2 Trusted Contexts

DB2 9.7 includes a 'trusted context' feature. Trusted contexts are defined by database objects that provide a specification for a trust relationship between the database and an external entity. The trust relationship is based on three attributes: an authorization name, a data stream encryption attribute (indicating whether the connection must be made across an encrypted channel – provided by the TOE's operational environment), and an IP address (or addresses). The user associated with any connection that matches the definition of a trusted context object is considered 'trusted' by the database.

Users trusted in this manner (e.g., 'trusted servers') can be configured such that they are allowed to modify some of the security attributes associated with their database connections. Specifically, they can be configured such that the 'user' associated with an existing connection to be changed depending on the DB2 configuration, this may or may not require authentication of the this user identity.

An additional feature of trusted contexts is role inheritance. A trusted context can define a default database role[3] to be granted to trusted context users or alternately a specific database role to be assigned to specific users associated with the trusted context. In either case, the trusted context user will be associated with the defined database role which could be in addition to any database roles assigned to that user (i.e., authorization name) directly.

This function is intended for multi-tier environments where the middle tier, typically an application server, might already perform authentication of end users. Trusted contexts provide a mechanism for the database to trust the middle tier and effectively establish connections on behalf of end users without necessarily supplying the credentials (password) of the end user to the database. Only a user with SECADM authority can configure this feature and as such it can be configured or not at the discretion of a DB2 security administrator.

2.2.4.3 SSL Client Connections

DB2 supports the use of SSL for java-based clients. As such, users of DB2 can choose to enable that feature though it is not necessary particularly when other means of client communication protection are configured in the operating environment of the TOE.

This feature is implemented using the IBM Global Security Kit (GSKit), which needs to be installed in conjunction with the DB2 product. The GSKit cryptographic services are utilized by the TOE in order to facilitate secure SSL client connections. The installation documentation for the TOE provides instructions for obtaining and installing the CC/FIPS certified version of the GSKit.

2.2.4.4 Authentication Servers

While DB2 requires that an authentication server is configured, users of DB2 can alternately configure DB2 to use authentication services of its underlying operating system, an externally available LDAP server, or an externally available KDC server. In each case DB2 relies upon that server for appropriate authentication services for its users.

2.2.4.5 User Credential Encryption

DB2 supports the ability to require that user IDs and passwords are encrypted by associated clients. By default this feature is not enabled, but can be enabled in environments where there may not be other protections adequate to protect the authentication information of users. For example, where an enterprise network may not be adequately protected or secure distributed operating system authentication mechanisms (such as within an operating system domain) may not be available. In this regard, DB2 supports the use of DES and AES-256. Given that DES is not a strong algorithm, involving only 56-bit keys and is no longer FIPS approved, AES-256 should be considered the preferred means of protection.

The AES-256 user ID and password protection feature is implemented using another IBM product -- the IBM Crypto for C (ICC) version 1.4.5 library linked with the TOE. Note that ICC v1.4.5 has been FIPS certified (certificate number 775), albeit in conjunction with earlier versions of the operating systems currently hosting DB2.

2.3 Product Features

The DB2 TOE offers the following features:

- Able to manage large volumes of data;
- Provides query and update ability via ANSI standard SQL;
- Provides rollback capability to preserve data integrity;
- Runs on multiple operating system and hardware platforms;

[3] The term 'database role' should not be confused with 'security role'. Database roles are a DB2-implemented construct similar in function to groups where privileges and authorities can be assigned to database roles to simplify their management when assigned to users. Security roles are used to represent the notion of security-relevant user roles to distinguish administrators and users with lesser authorities.

- DB2 provides support for both local and remote DB2 clients.

2.4 Security Environment TOE Boundary

The TOE includes both physical and logical boundaries.

2.4.1 Physical Boundaries

DB2 is physically a software application instantiated within the context of a host operating system, specifically IBM DB2 Enterprise Server Edition Version 9.7 for Linux, Unix, and Windows (with or without the IBM Base Warehouse Feature for DB2 9.7 license option). While DB2 can alternately be installed using a non-root install option, that configuration limits the functions of DB2 and has not been subject to evaluation.

As an application, the interfaces of DB2 are primarily logical in nature. The physical boundaries of the TOE are essentially the interfaces implemented by DB2. These interfaces can be divided into two general categories: interfaces to clients and interfaces to the hosting operational environment.

DB2 interacts with clients using the DRDA standard described previously. However, while the product is shipped with libraries and programs that expose other APIs (command line, ODBC, JDBC, etc.), the libraries and programs simply serve to convert their exposed APIs to DRDA flows to the DB2 server. With the exception of those tools and utilities identified in the TOE's guidance documentation, these libraries and programs are not security relevant (i.e., outside the TSF) and hence have not been specifically analyzed as part of the evaluation.

DB2 interacts with its operational environment using hardware instructions, operating system calls, and network protocols, just like all other applications. DB2 uses services of the hosting operational environment to instantiate itself as a set of executing processes, to store and retrieve data, to interact with remote clients, and to authenticate users.

The following components are required in the operational environment:

- **Hosting OS:** AIX 6, Linux RHEL 5 update 2, Linux SLES 10 with SP2, Microsoft Windows Server 2003 Enterprise Edition with SP2, or Solaris 10.
- **Authentication server (when not using the services of the hosting OS):** Any standard-compliant LDAP or KDC server.
- IBM Global Security Kit

2.4.2 Logical Boundaries

The logical boundaries of DB2 are realized in the security functions that it implements.

Note that the following subsections represent the security functions of DB2 that have been subject to evaluation. While the entire DB2 product is included within the TOE (except for a non-root install configuration as indicated above), not all functions shipped with the product have been subject to evaluation. In addition to non-security functions, the following security-related functions have not been subject to evaluation (i.e., there are no corresponding security claims):

- **DES:** While the TOE can be configured to use DES to protect authentication credentials, this mechanism has not been subject to evaluation and as such should not be solely relied upon as an adequate means of protection.
- **Data encryption functions (ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR, and GETHINT)[4]:** While users can employ these functions to encrypt and decrypt data, they have not been subject to evaluation and as such should not be solely relied upon as an adequate means of protection.
- **Unfenced routines:** Fenced routines execute in processes separate from the DB2 server, while unfenced routines share the DB2 server process (see section 6.1.5). Given that the DB2 server must protect itself (e.g., from tampering) and its ability to do so is limited when users can create routines that execute within

[4] Note that this is intended to address the functions shipped with the product, but any such functions developed by end users would also not be included within the scope of evaluation.

the same operating system process, unfenced routines are excluded from the evaluated configuration of the TOE.

- **CLIENT authentication:** The TOE supports a number of authentication configurations. While for the most part the TOE administrator can choose the configuration that best fits their specific environment, the configuration whereby the client is trusted to authenticate the user is excluded from the evaluated configuration. The evaluation address the configuration where the DB2 server is responsible to ensure that users are authenticated, although it relies on other configured components to do so.

2.4.2.1 Security Audit

DB2 records security relevant events that occur within its scope of control. These events are associated with individual users for individual accountability and can be accessed only by authorized administrators[5].

The audit log files, for DB2 instances and databases, are stored in files in the TOE's operational environment (i.e., underlying OS) configured during installation and the audit configuration file (db2audit.cfg) is located in each instance's security subdirectory. In addition to relying on the underlying OS to store and protect audit data stored in files, DB2 relies on the OS to provide reliable time information to record in its audit records.

2.4.2.2 Access Control

DB2 associates privileges and authorities with each individual user, group of users, and database role. These privileges and authorities are associated with operations that can be performed on the objects (e.g., database) that are implemented by DB2. DB2 uses identities, privileges, authorities, and access control lists associated with users, groups, roles, and objects to determine whether specific operations will be allowed when attempted by client users.

Note that while the term 'security roles' is used in this ST to distinguish authorized administrators from non-administrator users, DB2 implements this concept using a variety of authorities and privileges. DB2 implements a number of authorities - SYSADM, SYSCTRL, SYSMON, SYSMAINT, DBADM, SECADM, SQLADM, WLMADM, ACCESSCTRL, and DATAACCESS – of which SYSADM, SECADM, ACCESSCTRL, and DATAACCESS are particularly security relevant. SYSADM authority makes a user a system administrator that can utilizes most utility functions and is designed to manage and maintain DB2 instances. SECADM authority makes a user a security administrator with responsibility for security management and has primary control over access to security-relevant functions, audit, the Label-Based Access Control (LBAC) policy and role assignments. DBADM authority makes a user a database administrator and provides some management functions for a given database, but does not grant access to the data nor the ability to control access to objects within the database. Those functions are assigned to the DATAACCESS and ACCESSCTRL authorities, respectively. References to the 'user' security role are implemented in DB2 as any combination of lesser privileges (such as having the UPDATE or INSERT privilege on a specific database table). The other authorities grant various monitoring and tuning capabilities related to DB2, but do not serve to offer access to user data, for example.

In addition to using privileges and authorities to control access, DB2 implements a LBAC mechanism. The DB2 security administrator can grant (or revoke) security labels and exemptions to (or from) users as well as create and drop LBAC security objects in order to define LBAC polices for specific database tables. Once a table is configured with a LBAC policy (i.e., the table is LBAC protected relative to either rows or columns), users must additionally satisfy the LBAC access rules in order to access or modify the applicable table rows or columns.

2.4.2.3 Identification & Authentication

DB2 requires all users to be identified and authenticated before allowing them access to DB2 resources. The TOE's operational environment (i.e., host operating system, LDAP server, or KDC server) performs the actual authentication and association of users with groups and passes the result to DB2. DB2 subsequently enforces the result returned by the TOE's operational environment and uses the user identity and group memberships (i.e., list of groups) returned by the TOE's operational environment, along with its own associations of users, groups, and other

[5] The term *authorized administrator* is used to generally refer to an administrator authorized (e.g., by authority or privilege) to perform a corresponding function depending on the context in which the term is used.

database roles with database roles, to associate privileges, authorities, and security labels and exemptions with the authenticated user.

Note that the association between users and groups is managed within the TOE's operational environment. Operational environment user and group identities are uniquely mapped in the TOE and when a user accesses the TOE, the operational environment provides the user and all group identities associated with that user. However, database roles are defined within the TOE where users, groups, and other database roles can be associated with specific database roles.

Note that servers defined as trusted via the trusted context feature must be authenticated like all other users. However, they have the ability to use alternate identities, without further authentication, in accordance with their definition in the TOE. They could be configured so that they can use only a specific set of identities or alternately so that they can use any identity known to the TOE.

2.4.2.4 Security Management

DB2 includes the security roles of system administrator, security administrator, and user (with various combinations of authorities and privileges), implemented using DB2 authorities and privileges, and allows individual users to be assigned to those security roles by virtue of group assignments in the TOE's operational environment. Management of the DB2 TOE, including the ability to select and review audit records, is restricted to appropriate administrators based on authorities and privileges. Management of DB2 objects, including management of security labels, as well as database roles and audit policies is restricted to those users that are assigned the appropriate privileges to do so.

Note that the users trusted by virtue of a trusted context configuration could have the ability to assert alternate identities without requiring authentication by the TOE. While not a security role per se, this feature has been subject to evaluation.

2.4.2.5 TOE Protection

DB2 communicates between DPF instances, when so configured, and also with clients that can be remote from the DB2 server. DB2 implements SSL, using a separate GSKit product in the TOE's operational environment for cryptographic services, which is available only to java-based clients. Otherwise, DB2 relies upon its operational environment to ensure adequate communication protections. In the case of DPF instances a dedicated network can be configured to be used exclusively by DB2. In the case of remote clients, if all the hosts on the applicable network are not adequately trusted, IPSec or other host- or network-based protection mechanisms could be configured to protect any otherwise insecure network traffic.

As summarized in section 2.2.4.5, DB2 also provides the optional feature to require that user IDs and passwords are encrypted by the associated user clients in order to offer more protection for user authentication data.

Note that DB2 also depends on its operational environment to provide a reliable source of time information.

2.4.2.5.1 Additional Architecture-based Protections

DB2 executes within processes provided and protected by the hosting operational environment. However, it is designed to not share its process space with non-TOE entities in order to ensure that TSF resources are protected. DB2 has been designed so that each of its interfaces performs the necessary access checks before allowing access to DB2 resources.

2.5 TOE Documentation

IBM provides administrator and user guidance on how to utilize the TOE security functions and warnings to authorized administrators and users about actions that can compromise the security of the TOE. The installation and generation procedures, included in the administrator guidance, describe the steps necessary to install DB2 in accordance with the evaluated configuration.

IBM also provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. IBM's delivery procedures describe the electronic and non-electronic procedures to be used to detect modification to the TOE.

All of the administrator and user guidance is documented in:

- IBM DB2 Common Criteria Certification: Administration and User Documentation
- IBM DB2 Common Criteria Certification: Installing DB2 Enterprise Server Edition
- IBM DB2 Delivery Procedures document

3. Security Problem Definition

Since DB2 was developed based largely on the Trusted Computer System Evaluation Criteria (TCSEC) C2 security requirements, the security environment has been modeled after that specified in the Controlled Access Protection Profile (CAPP), which is the successor to TCSEC C2 in the context of the Common Criteria (CC). Note, however, that since DB2 is a database system and not an operating system, some additional assumptions and security objectives have been assigned to the operational environment of the TOE.

3.1 Secure Usage Assumptions

The usage assumptions are organized in three categories: personnel (assumptions about administrators and users of the system as well as any threat agents), physical (assumptions about the physical location of the TOE or any attached peripheral devices), and connectivity (assumptions about other IT systems that are necessary for the secure operation of the TOE).

3.1.1 Personnel Assumptions

It is assumed that the following personnel conditions will exist:

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.NO_EVIL_ADM

The administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.COOP

Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

A.CLEARANCE

Procedures exist for granting users authorization for access to specific security levels. It is further assumed the TOE administrators will be cleared to the highest security level processed by the TOE.

3.1.2 Physical Assumptions

The TOE is intended for application in areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

A.LOCATE

The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

A.PROTECT

The hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

3.1.3 Connectivity Assumptions

It is assumed that the following connectivity conditions exist:

A.CONNECT

All connections to peripheral devices reside within the controlled access facilities. The TOE only addresses security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths*, including networks, to access points and between TOE instances* are assumed to be adequately protected.

A.PLATFORM

The operational environment underlying the TOE is assumed to fulfill the objectives for the TOE-supporting components in the operational environment described in this ST.

3.2 Threats

All TOE and environment security objectives have been derived from the statement of Organizational Security Policy or Secure Usage Assumptions found in the following sections. Therefore, there is no statement of the explicit threats countered by the TOE.

3.3 Organization Security Policies

An Organizational Security Policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. Although some of the organizational security policies described below are drawn from the CAPP they apply to many non-DoD environments.

P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the TOE may access the TOE.

P.NEED_TO_KNOW

The TOE must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a “need to know” for that information.

P.ACCOUNTABILITY

The users of the TOE can be held be accountable for their actions within the TOE.

P.CLASSIFICATION

The system must be able to limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at.

4. Security Objectives

This section defines the security objectives of the TSF and its supporting environment. Security objectives, categorized as either applying to the TOE or its environment, reflect the stated intent to comply with any assumptions and organizational security policies identified. All of the identified assumptions and organizational security policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

O.DISCRETIONARY_ACCESS

The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which users may access which resources.

O.MANDATORY_ACCESS

The TSF must be able to control access to resources based upon the sensitivity and categories of the information being accessed and the clearance of the subject attempting to access that information.

O.AUDITING

The TSF must be able to record the security relevant actions of users of the TOE. The TSF must be able to present this information only to authorized administrators.

O.RESIDUAL_INFORMATION

The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

O.ROLLBACK

The TSF must ensure that operations performed on information contained in a protected resource can be undone before the results have been committed.

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

O.ENFORCEMENT

The TSF must be designed and implemented in a manner that ensures that the organizational policies are enforced in the target environment.

4.2 Security Objectives for the Operational Environment

The TOE is assumed to be complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. The following are the security objectives for the environment:

4.2.1 Ojectives for the use of the TOE in its operational environment

O.ADMIN_GUIDANCE

Appropriate guidance documentation must be provided to enable administrators to install, manage, and operate the TOE in a manner that maintains IT security objectives.

O.ADMINISTRATORS

Administrators of the TOE and its operational Environment must not be careless, willfully negligent or hostile, and must follow the instructions provided in the administrator guidance documentation.

O.ASSIGN

One or more competent individuals must be assigned to manage the TOE and the security of the information it contains.

O.COOP

Authorized users must possess the appropriate authorization to access at least some of the information managed by the TOE and must act in a cooperative manner in a benign environment.

O.INSTALL

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner, which maintains IT security objectives.

O.PHYSICAL

Those responsible for the TOE must ensure that those parts of the physical TOE and its associated operational environment critical to security policy are protected from attack, which might compromise IT security objectives.

O.CREDEN

Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner that maintains IT security objectives and that credentials (e.g., clearances) are assigned appropriately, including ensuring that administrators are cleared to the highest security level processed by the TOE.

O.PLATFORM

Those responsible for the TOE must ensure that the components underlying the TOE fulfill the objectives for its operational environment described in this ST.

4.2.2 Objectives for TOE-supporting components in the operational environment

OE.AUTHORIZATION

The TOE's operational environment must ensure that only authorized users gain access to the operational environment and its resources. The operational environment must support the TOE by ensuring that users are adequately authenticated on the TOE's behalf.

OE.AUDITING

The TOE's operational environment must be able to record the security relevant actions of users of the operational environment.

OE.CRYPTO

The TOE's operational environment must provide cryptographic services suitable to allow the TOE to establish secure SSL connections.

OE.RESIDUAL_INFORMATION

The TOE's operational environment must ensure that any information contained in a protected resource that may be assigned to the TOE is not released when the resource is recycled.

OE.MANAGE

The TOE's operational environment must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of operational environment security, including security relevant support for the TOE.

OE.ENFORCEMENT

The TOE’s operational environment must be designed and implemented in a manner that ensures that it can protect the operational Environment of the TOE. The TOE’s operational environment must provide a reliable time source and secure audit storage for the use of both the TOE and its operational Environment.

5. IT Security Requirements

The following sections define the security functional and assurance requirements for the TOE and its operational environment. The security functional requirements have been drawn largely from the Controlled Access Protection Profile (CAPP) and the security assurance requirements have been drawn from EAL 4, as defined in the CC Part 3, augmented with ALC_FLR.1.

5.1 Extended Components Definition

This Security Target includes two extended security functional requirements (FIA_UAU_EXP.2 and FPT_STM_EXP.1) and no extended security assurance requirements. See Sections 8.5, 5.2.3.2, and 5.2.5.1 for the component descriptions and rationale.

5.2 TOE Security Functional Requirements

This section specifies the security functional requirements that are applicable to the TOE.

Table 1 TOE Functional Security Requirements

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SAR.3: Selectable audit review
	FAU_SEL.1: Selective audit
	FAU_STG.3: Action in case of possible audit data loss
	FAU_STG.4: Prevention of audit data loss
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
	FDP_IFC.1: Subset information flow control
	FDP_IFF.2: Hierarchical security attributes
	FDP_RIP.2: Full residual information protection
	FDP_ROL.1: Basic rollback
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU_EXP.2: User authentication before any action
	FIA_UID.2: User identification before any action
	FIA_USB.1: User-subject binding
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1a: Management of security attributes
	FMT_MSA.1b: Management of security attributes
	FMT_MSA.3a: Static attribute initialization
	FMT_MSA.3b: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_REV.1: Revocation
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_STM_EXP.1: Reliable time stamps
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel

5.2.1 Security audit (FAU)

5.2.1.1 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 the [***not specified***] level of audit; and

c) [**the audit events identified in Table 2 Auditable 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, [**the additional details identified in Table 2 Auditable Events**].

Table 2 Auditable Events for the TOE

Component	Event	Additional Details
FAU_GEN.1	Start-up and shutdown of the audit functions	
FAU_SAR.1	Reading of information from the audit records	
FAU_SAR.2	Unsuccessful attempts to read information from the audit records.	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating	
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP.	Object name
FDP_IFC.1	All requests to perform an operation on an object covered by the SFP.	Object name
FIA_UAU_EXP.2	All use of the authentication mechanism	
FIA_UID.2	All use of the user identification mechanism, including the identity provided during successful attempts	
FIA_USB.1	Success and failure of binding user security attributes to a subject (e.g. success and failure to create a subject)	
FMT_MOF.1	All modifications in the behaviour of the functions in the TSF	Command issued
FMT_MSA.1a	All modifications of the values of security attributes	Command issued
FMT_MSA.1b	All modifications of the values of security attributes	Command issued
FMT_MTD.1a	All modifications to the values of TSF data	Command issued
FMT_MTD.1b	All modifications to the values of TSF data	Command issued
FMT_MTD.1c	All modifications to the values of TSF data	Command issued
FMT_REV.1	All modifications to the values of TSF data	Command issued
FMT_SMF.1	Use of the management functions	Command issued

5.2.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**system administrators, security administrators, or users granted privilege to execute audit routines**] with the capability to read **and archive** [**audit information as specified in the table below**] from the audit records.

Role	Audit Information

System Administrator	All audit data for the DB2 instance
Security Administrator	All audit data for the databases to which they are assigned
User granted privilege to execute audit routines	All audit data for the databases to which they have been granted access

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.1.4 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.2.1.5 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to apply [**searches**] of audit data based on [**user identity**].

5.2.1.6 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to select the set of audited events from the set of all auditable events based on the following attributes:

a) [***event type, user identity,***]

b) [**group membership, database role, database instance, database table, authority, and success and/or failure**].

5.2.1.7 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall [**write an entry in a separate administrator log**] if the audit trail exceeds [**the available space allocated to the audit log**].

5.2.1.8 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall [***'ignore audited events'* or *'prevent audited events, except those taken by the authorised user with special rights'***] and [**no other action**] if the audit trail is full.

Application Note: The system administrator must choose to configure DB2 to either discard audit events or to effectively stop once the audit trail is full.

5.2.2 User data protection (FDP)

5.2.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] on [**user attempts to create, destroy or otherwise access databases, schemas, table spaces, tables, views, packages, procedures, functions, and methods**].

5.2.2.2 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: [**subject and object attributes as defined in the table below**].

Controlled entity	Security attributes
Subjects	
User	Authorization names and authorities

Objects	
Database Table space	Access control list[6]
Schema Table View Package Procedure Function Method	Access control list Owner

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**a subject must have an authorization name that is assigned the privilege (per the access control list) corresponding to the requested operation of the target object in order to succeed in performing the requested operation**].

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

1) **a subject that is assigned the DATAACCESS authority to a database can access objects of that database as allowed by the authority regardless of privileges (per the access control list) and**

2) **a subject that has an authorization name that is the owner of the applicable object can access the object regardless of privileges (per the access control list)**].

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

5.2.2.3 Subset information flow control (FDP_IFC.1)

FDP_IFC.1.1 The TSF shall enforce the [**LBAC SFP**] on [**user read and write operations on LBAC protected database tables**].

5.2.2.4 Hierarchical security attributes (FDP_IFF.2)

FDP_IFF.2.1 The TSF shall enforce the [**LBAC SFP**] based on the following types of subject and information security attributes: [**user security labels and database table column or row security labels**].

Application Note:
Note that security labels consist of zero (0) or more of each of the three (3) available component types (array, set, and tree), but must include at least one component.

- Array – represents an ordered set; any element in the set is ranked higher than subsequent elements in the set.
- Set – represents an unordered set; there is no defined relationship among the elements in the set and there order is not important.
- Tree – represents a hierarchy and is used to represent organizational charts and to identify departments within an organization that owns the applicable data. An element of a tree that is higher than another element in the tree hierarchy is considered an *ancestor*.

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, based on the ordering relationships between security attributes hold: [

1) **in order to read a LBAC protected column or row in a database table:**

a) the array components of the user's security label must be greater than or equal to the array components of the object's security label,

b) the set components of the user's security label must include the set components of the object's security label, and

[6] Access control lists assign privileges to users via authorization names.

c) the tree components of the user's security label must include at least one of the elements in the tree components of the object's security label (or the ancestor of one such element) and

2) in order to write a LBAC protected column or row in a database table:

a) the array components of the user's security label must be equal to the array components of the object's security label,

b) the set components of the user's security label must include the set components of the object's security label, and

c) the tree components of the user's security label must include at least one of the elements in the tree components of the object's security label (or the ancestor of one such element) and

3) the Discretionary Access Control Policy rules must be satisfied in every case].

FDP_IFF.2.3 The TSF shall enforce the [**rule that only a security administrator can change security labels on users and an appropriately privileged user can change security labels on columns or rows of LBAC protected tables**].

FDP_IFF.2.4 The TSF shall explicitly authorise an information flow based on the following rules: [**a user with the appropriate corresponding exemption can ignore the read array, read set, read tree, write array (to lower array values), write array (to higher array values), write set, or write tree check**].

FDP_IFF.2.5 The TSF shall explicitly deny an information flow based on the following rules: [**none**].

FDP_IFF.2.6 The TSF shall enforce the following relationships for any two valid information flow control security attributes: a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable; and b) There exists a 'least upper bound' in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and c) There exists a 'greatest lower bound' in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

5.2.2.5 Full residual information protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.2.2.6 Basic rollback (FDP_ROL.1)

FDP_ROL.1.1 The TSF shall enforce **the** [**Discretionary Access Control Policy and LBAC SFP**] to permit the rollback of the [**operations that can be expressed as SQL**] on the [**databases, schemas, tables spaces, tables, views, packages, procedures, functions, and methods**].

FDP_ROL.1.2 The TSF shall permit operations to be rolled back within the [**set of uncommitted statements**].

5.2.3 Identification and authentication (FIA)

5.2.3.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**authorization names**, **authorities, database roles, security labels, exemptions, and in the case of trusted contexts the trusted context object including a list of authorized authorization names, database roles, encryption attribute, and IP addresses**].

5.2.3.2 User authentication before any action (FIA_UAU_EXP.2)

FIA_UAU_EXP.2.1 The TSF shall require each user to be successfully authenticated using support from its environment before allowing any other TSF-mediated actions on behalf of that user.

5.2.3.3 User identification before any action (FIA_UID.2)

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

5.2.3.4 User-subject binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**authorization name**, **authorities, database roles, security labels, exemptions, and in the case of trusted contexts the trusted context object and IP address of the trusted user connection**].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**when user is successfully identified and authenticated, authorization names associated with the user, the user's groups, and the roles to which those user, groups, or other roles belong are assigned to the session as are authorities, security labels, and exemptions associated with those authorization names; additionally, in the case of trusted contexts if the trusted context definition defines a database role for the authorization name, that database role is assigned to the resulting session ELSE if the trusted context definition defines a default database role, that database role is assigned to the resulting session**].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [**once a session is created its attributes do not change except as follows:**

a user in the trusted context role can change the authorization name of any of its connections to any authorization name authorized for that trusted context

- – **depending on TOE configuration a password could be required for authentication of the new authorization name AND**
- – **once the authorization name is changed FIA_USB.1.2 is applied to the connection as though it is an initial association (including the association of authorities, roles, security labels, and exemptions associated with the new authorization name) AND**
- – **if the trusted context definition defines a database role for the new authorization name, that database role is assigned to the resulting session ELSE if the trusted context definition defines a default database role, that database role is assigned to the resulting session**].

5.2.4 Security management (FMT)

5.2.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behaviour of***] the functions [**LBAC SFP**] to [**the security administrator**].

5.2.4.2 Management of security attributes (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to restrict the ability to [***modify***] the security attributes [**specifically, access control attributes, associated with a protected object and database role assignments**] to [**users authorized by the Discretionary Access Control rules and users with the ACCESSCTRL authority for the applicable database**].

Application Note:
Note that the Discretionary Access Control rules encompass authorities. While access control attributes and role assignments can be granted and revoked by users with appropriate privileges, users with the ACCESSCTRL authority can always change access privileges within their assigned database and security administrators can always grant and revoke database roles.

5.2.4.3 Management of security attributes (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the [**LBAC SFP**] to restrict the ability to [***[assign]***] the security attributes [**security labels**] to [**users authorized by the LBAC Rules**].

Application Note:
Note that the LBAC policy allows users to assign labels to objects depending on their privileges, but labels can be granted or revoked t and from users only by a security administrator.

5.2.4.4 Static attribute initialization (FMT_MSA.3a)

FMT_MSA.3a.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~~ **Discretionary Access Control Policy**.

FMT_MSA.3a.2 The TSF shall allow the [**no role**] to specify alternative initial values to override the default values when an object or information is created.

5.2.4.5 Static attribute initialization (FMT_MSA.3b)

FMT_MSA.3b.1 The TSF shall enforce the [**LBAC SFP**] to provide [***[no]***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3b.2 The TSF shall allow the [**users authorized by the LBAC Rules**] to specify alternative initial values to override the default values when an object or information is created.

5.2.4.6 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to [***delete* and *[create]***] the [**audit trail**] to [**system administrators**].

5.2.4.7 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to [***modify* and *[observe]***] the [**set of audited events**] to [**security and system administrators**].

5.2.4.8 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to [***[create and drop]***] the [**roles**] to [**security administrators**].

5.2.4.9 Revocation (FMT_REV.1)

FMT_REV.1.1 The TSF shall restrict the ability to revoke [**Discretionary Access Control and LBAC security attributes**] associated with the [***objects***] under the control of the TSF to [**users authorised to modify the Discretionary Access Control security attributes by the Discretionary Access Control policy (including those granted the ACCESSCTRL authority for the applicable database and security administrators for the applicable database) and security administrators in the case of LBAC security attributes**].

Application Note:
Note that the Discretionary Access Control rules encompass authorities. While access control attributes can be granted and revoked by users with appropriate privileges, users with the ACCESSCTRL authority can always change access privileges within their assign database as can security administrators who can also grant and revoke database roles.

FMT_REV.1.2 The TSF shall enforce the rules [**the access rights associated with an object shall be enforced when an access check is made**].

5.2.4.10 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [
start and stop auditing;
select audited events;

create, delete, and review the audit trail;
create and drop LBAC policies and labels;
grant and revoke LBAC security labels and exemptions;
create, drop, grant, and revoke database roles; and
grant and revoke DAC access attributes].

5.2.4.11 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**system administrator, security administrator, and user (authorized by the DAC or LBAC rules, with ACCESSCTRL authority, or granted privilege to execute audit routines)**].

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

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Reliable time stamps (FPT_STM_EXP.1)

FPT_STM_EXP.1.1 The TSF shall be able to provide reliable time stamps based on information provided by its environment for its own use.

5.2.6 Trusted path/channels (FTP)

5.2.6.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [***another trusted IT product***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**no required functions**].

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 (EAL 4) components as specified in Part 3 of the Common Criteria, augmented with ALC_FLR.1 as indicated in bold the following table. No operations are applied to the assurance components.

Table 3 Assurance Requirements (EAL 4 augmented)

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.4: Complete functional specification
	ADV_IMP.1: Implementation representation of the TSF
	ADV_TDS.3: Basic modular design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.4: Production support, acceptance procedures and automation
	ALC_CMS.4: Problem tracking CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_FLR.1: Basic flaw remediation
	ALC_LCD.1: Developer defined life-cycle model

Requirement Class	Requirement Component
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.2: Testing: security enforcing modules
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3: Focused vulnerability analysis

5.3.1 Development (ADV)

5.3.1.1 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 initialisation 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.

5.3.1.2 Complete functional specification (ADV_FSP.4)

ADV_FSP.4.1d The developer shall provide a functional specification.
ADV_FSP.4.2d The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.4.1c The functional specification shall completely represent the TSF.
ADV_FSP.4.2c The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.4.3c The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.4.4c The functional specification shall describe all actions associated with each TSFI.
ADV_FSP.4.5c The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.
ADV_FSP.4.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.4.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.1.3 Implementation representation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall make available the implementation representation for the entire TSF.
ADV_IMP.1.2d The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.
ADV_IMP.1.1c The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2c The implementation representation shall be in the form used by the development personnel.
ADV_IMP.1.3c The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

ADV_IMP.1.1e The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

5.3.1.4 Basic modular design (ADV_TDS.3)

ADV_TDS.3.1d The developer shall provide the design of the TOE.

ADV_TDS.3.2d 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.3.1c The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2c The design shall describe the TSF in terms of modules.

ADV_TDS.3.3c The design shall identify all subsystems of the TSF.

ADV_TDS.3.4c The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5c The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6c The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7c The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.

ADV_TDS.3.8c The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.

ADV_TDS.3.9c The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10c The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

ADV_TDS.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.3.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance documents (AGD)

5.3.2.1 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 fulfil 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.

5.3.2.2 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.

5.3.3 Life-cycle support (ALC)

5.3.3.1 Production support, acceptance procedures and automation (ALC_CMC.4)

ALC_CMC.4.1d The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.4.2d The developer shall provide the CM documentation.

ALC_CMC.4.1c The TOE shall be labelled with its unique reference.

ALC_CMC.4.2c The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3c The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4c The CM system shall provide automated measures such that only authorised changes are made to the configuration items.

ALC_CMC.4.5c The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6c The CM documentation shall include a CM plan.

ALC_CMC.4.7c The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8c The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9c The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10c The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMC.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.2 Problem tracking CM coverage (ALC_CMS.4)

ALC_CMS.4.1d The developer shall provide a configuration list for the TOE.

ALC_CMS.4.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2c The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.3 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.

5.3.3.4 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.

5.3.3.5 Basic flaw remediation (ALC_FLR.1)

ALC_FLR.1.1d The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2c 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.1.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4c 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.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.6 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.

5.3.3.7 Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of each development tool.

ALC_TAT.1.1c Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3c The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

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

5.3.4 Tests (ATE)

5.3.4.1 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.

5.3.4.2 Testing: security enforcing modules (ATE_DPT.2)

ATE_DPT.2.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and SFR-enforcing modules in the TOE design.

ATE_DPT.2.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.2.3c The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.

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

5.3.4.3 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. These scenarios shall include any ordering dependencies on the results of other tests.

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.

5.3.4.4 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.

5.3.5 Vulnerability assessment (AVA)

5.3.5.1 Focused vulnerability analysis (AVA_VAN.3)

AVA_VAN.3.1d The developer shall provide the TOE for testing.

AVA_VAN.3.1c The TOE shall be suitable for testing.

AVA_VAN.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.3.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.3.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.3.4e 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 Enhanced-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 DB2 audit facility acts both at an instance level, recording all instance level activities, and at the database level for database specific activities. The audit log files, for instances and databases, are stored in file locations configured during installation and the audit configuration file (db2audit.cfg) is located in each instance's security subdirectory.

Users of the audit facility administrator tool, db2audit, must have SYSADM authority/privileges (i.e., they must be a system administrator). The audit facility must be stopped and started explicitly, within the context of an instance, by a system administrator using db2audit which will perform its function only if the user has the SYSADM authority. When starting, the audit facility uses the instance's existing audit configuration information. Since the audit facility is independent of the DB2 server (i.e., it runs in a separate OS-provided process), it will remain active even if the corresponding instance is stopped. In fact, when the instance is stopped, an audit record may be generated in the instance's audit log.

System administrators using the audit facility tool can control the following actions within the audit facility:

- Start recording auditable events within the DB2 instance.
- Stop recording auditable events within the DB2 instance.
- Flush any pending audit records from the instance and write them to the audit log.
- Configure the behavior of the audit facility, including selecting the categories of the auditable events to be recorded.
- Configure whether the audit facility should prevent auditable events or ignore auditable events in the event that the audit log becomes full.
- Request a description of the current audit configuration.
- Extract audit records by formatting and copying them from the audit log to a flat file or ASCII delimited files. Extraction is done in preparation for analysis of log records.
- Archive the current audit log for either an individual database or the instance to a new location for archiving and later extraction. The current timestamp will be appended to the filename. All records that are currently being written to the audit log will complete before the log is archived to ensure full records are not split apart. All records that are created while the archive is in progress will be written to the current audit log, and not the archived log, once the archive has finished.

Note that these commands serve to manipulate the audit service directly, manage the audit configuration file, and manage the audit log as applicable.

When a DB2 instance records audit records, they contain the following information although some of the audit categories will contain more:

- Timestamp – date and time of the audit event
- Category – general type for the audit event
- Audit Event – specific audit event identifier
- Event Correlator – correlation identifier for the operation audited (Can be used to associate multiple records resulting from a single event.)
- Event Status – success or failure of the event. Unsuccessful audit events are represented by a SQLCODE.

- User ID – user identifier associated with the audit event
- Authorization ID – authorization identifier (or name) associated with the audit record

In addition, object names associated with access and information flow decisions and the specific commands issued to perform security management functions are identified in applicable audit records.

DB2 provides the system administrator (for instance level audit logs), security administrator (for database level audit logs), or a user granted, by a security administrator, execute privilege to the applicable audit routines (for database level audit logs) with a capability to search the audit records by user identity in order to ensure accountability of actions to the appropriate individual. To achieve this, audit records can be extracted, by the authorized user, in ASCII format and loaded into a DB2 relational table for a rich set of query capabilities.

The system administrator can configure DB2 to audit any or all of the available audit categories for instance level auditing:

- Audit (AUDIT) – Generates records when audit settings are changed or when the audit log is accessed.
- Statement Execution (EXECUTE) – Generates records of just the SQL statement that is being executed. Note that the CONTEXT audit category can be used to audit the use of statements, but it captures much more information which may not be necessary for a given application.
- Authorization Checking (CHECKING) – Generates records during authorization checking of attempts to access, transfer, or manipulate DB2 objects or functions.
- Object Maintenance (OBJMAINT) – Generates records when creating or dropping data objects.
- Security Maintenance (SECMAINT) – Generates records when granting or revoking: object or database privileges or authorities, database roles, security labels, or (LBAC) exemptions, or transfer ownership of objects Records are also generated when the database manager security configuration parameters SYSADM_GROUP, SYSCTRL_GROUP, SYSMON_GROUP, or SYSMAINT_GROUP are modified.
- System Administration (SYSADMIN) – Generates records when operations requiring SYSADM, SYSMAINT, or SYSCTRL authority are performed.
- User Validation (VALIDATE) – Generates records when authenticating users or retrieving system security information.
- Operation Context (CONTEXT) – Generates records to show the operation context when a database operation is performed. This category allows for better interpretation of the audit log file. When used with the log's event correlator field, a group of events can be associated back to a single database operation. For example, an SQL statement for dynamic SQL, a package identifier for static SQL, or an indicator of the type of operation being performed, such as CONNECT, can provide needed context when analyzing audit results.

In addition, the security administrator (i.e., SECADM) can configure DB2 to make use of audit policies to audit events based on categories - identifying whether they should never or always be audited or just when they succeed or fail. The audit policies can be associated with specific users, groups of users, database roles, authorities (SYSADM, SYSCTRL, SYSMAINT, SYSMON, DBADM, SECADM, SQLADM, WLMADM, DATAACCESS, and ACCESSCTRL), database tables, and database instances so that audit selection can be fined-tuned to target the success and/or failure of specific events, user, groups, database roles, authorities, database tables, and database instances at the discretion of the security administrator.

DB2 does not provide any ability within the TOE to modify the audit records. Due to role restrictions, DB2 offers functions allowing only a system administrator or user granted execute privilege, by a security administrator, on the applicable audit routine to delete stored audit records. Should the audit trail exceed a pre-defined limit (the available space on the file system containing the audit log) a message is inserted in the administrator's log, which is not part of the audit log. Furthermore, a system administrator can configure DB2 to stop auditing or stop the current SQL statement or other auditable event (effectively preventing auditable events) when the audit trail becomes full.

The Security Audit security function satisfies the following security requirements:

FAU_GEN.1 Audit data generation – DB2 fulfills this requirement by generating the necessary events associated with each of its security functions (and security functional requirements) and by including the date and time, event type, user and authorization identities, and results in each event along with object names and security management commands where applicable.

FAU_GEN.2 User identity association – DB2 fulfills this requirement by including the applicable user identity in each audit record.

FAU_SAR.1 Audit review – DB2 fulfills this requirement by providing interfaces to the system administrator, applicable security administrator, and users granted execute access to the applicable audit routines for the review and archival of audit records.

FAU_SAR.2 Restricted audit review – DB2 fulfills this requirement by ensuring that the user is a system administrator or security administrator (per their role) or alternately has been granted execute privilege to the applicable audit routines before allowing access to the audit records associated with their role or privileges.

FAU_SAR.3 Selectable audit review – DB2 fulfills this requirement by providing search capabilities that can be realized by first exporting the audit trail and then importing back into a database where arbitrary queries could be made.

FAU_SEL.1 Selective audit – DB2 fulfills this requirement by allowing system and security (see above) administrators to configure DB2 to audit any or all of the available audit categories as well as successful and/or failed events, specific user identities, groups, database roles, authorities, database tables, and database instances.

FAU_STG.3 Action in case of possible audit data loss – DB2 fulfills this requirement by writing a record in the administrator log indicating when the audit trail is full.

FAU_STG.4 Prevention of audit data loss – DB2 fulfills this requirement by allowing a system administrator to configure DB2 to either simply throw away new audit events (i.e., stop auditing) or to stop processing auditable SQL commands when the audit trail becomes full.

6.1.2 Access Control

Authorization (see Section 6.1.3) is the process whereby DB2 obtains information about an authenticated DB2 user, indicating the database operations that user may perform, and what data objects may be accessed. With each user request, there may be more than one authorization check, depending on the objects and operations involved.

DB2 logically associates access control lists and an 'owner'[7] with each object using tables and configuration files to record the access permissions associated with each authorization name. The authorization name of an authenticated user, and those of groups and database roles to which the user belongs, are compared with access control list entries to find matches. Based on this comparison, DB2 identifies available permissions that indicate whether to allow the requested access.

There are two types of permissions managed by DB2: privileges and authority levels. A privilege defines a single permission for an authorization name, enabling a user to create or access database resources. Privileges are stored in the database catalogs. Authority levels provide a method of grouping privileges and control over higher-level database manager maintenance and utility operations. Database-specific authorities are stored in the database catalogs. Both privileges and database authorities can be associated with group and database role memberships. System authorities are associated with group memberships, and are stored in the database manager configuration file for a given instance.

Groups and database roles provide convenient means of performing authorization for a collection of users without having to grant or revoke privileges for each user individually. Unless otherwise specified, group and database role authorization names can be used anywhere that authorization names are used for authorization purposes. In general, group membership is considered for dynamic SQL and non-database object authorizations (such as instance level

[7] Note that a owner is associated only with the following database objects: Schema, Table, View, Package, Procedure, Function, and Method.

commands and utilities), but while group membership is not considered for static SQL database role membership is applicable. The exception to this general case occurs when privileges are granted to PUBLIC: these are considered when static SQL is processed.

Information about each database is automatically maintained in a set of views called the system catalog, which is created when the database is generated. This system catalog describes tables, columns, indexes, programs, privileges, and other objects and some of their attributes.

DB2 supports a number of Specific Privileges. They are:

- Database privileges, which involve actions on a database as a whole.
- Schema privileges, which involve actions on schemas in a database.
- Table space privileges, which involve actions on table spaces
- Table and view privileges, which involve actions on tables or views in a database.
- Package privileges, which involve actions on packages.
- Procedure, function, and method privileges, which involve actions on routines such as functions, procedures and methods.

In order for a user to access (including create and destroy) any of the objects mentioned in conjunction with the privileges identified above (i.e., databases, schemas, table spaces, tables, views, packages, procedures, functions, or methods), the user must be assigned the privilege corresponding with the action they are attempting to perform. Otherwise, the operation will be denied.

Note that authorities can be used to access objects without explicitly having the necessary privilege. The ACCESSCTLR authority, for example, permits the user to change access permissions on objects within the applicable database. Similarly, the DATAACCESS authority permits access to the data stored within objects of the applicable database. Furthermore, an object owner is not subject the privilege restrictions on the objects they defined (i.e., created).

Authorized administrators (i.e., users granted applicable authorities and privileges) have the ability to grant authorities, privileges, and database roles (associated with their specific administrative role) to other users, groups, and database roles. The administrator may optionally grant a privilege to a user WITH GRANT OPTION. Non-administrator users who hold a privilege WITH GRANT OPTION have the ability to grant that privilege to (but not to revoke that privilege from) other non-administrator users. The administrator may optionally grant a database role to a user WITH ADMIN OPTION. This option would grant the user the authority to grant the database role to other users.

In addition to controlling access using permissions, when a LBAC is properly configured, security administrators can define LBAC security labels and authorized users can assign LBAC policies to tables. Once a LBAC policy is assigned to a table (the notion of ‘protecting’ a table), if the table contains a column of type ‘DB2SECURITYLABEL’ the table is protected with row level granularity, also if the table has a column protected with a security label (per the table definition) the table is protected with column level granularity. Subsequently, when a user attempts to create, modify, or otherwise access data in the table their access is restricted, in addition to the Discretionary Access Control rules, based on the security label associated with their session, the security label(s) associated with the table, and the LBAC access rules. Hence, the requested access to specific rows or columns is subject to the LBAC constraints. Note that if a table is protected with both column- and row-level granularity, first the column check must succeed and then each applicable row check must succeed.

Note that when a new row is introduced in a table, if the table is protected with row level granularity, the user may specify a security label for that row in accordance with the LBAC rules. If they do not explicitly specify a security label, the user’s security label is used.

LBAC labels have zero (0) or more of each of the three available component types (but must always have at least one component):

> **Array** – represents an ordered set; any element in the set is ranked higher than subsequent elements in the set.

Set – represents an unordered set; there is no defined relationship among the elements in the set and there order is not important.

Tree – represents a hierarchy and is used to represent organizational charts and to identify departments within an organization that owns the applicable data.

There are two sets of three rules that determine the allowed access based on LBAC labels:

Read Access Rules apply when data is retrieved. Data is retrieved during SELECT, UPDATE, and DELETE operations.

DB2LBACREADARRAY – Each array component of the user's security label must be greater than or equal to the corresponding array component of the data (row or column) security label.

DB2LBACREADTREE – Each tree component of the user's security label must include at least one of the elements in the corresponding tree component of the data (row or column) security label (or the ancestor of one such element).

DB2LBACREADSET – Each set component of the user's security label must include the corresponding set component of the data (row or column) security label.

Write Access Rules apply for INSERT, UPDATE, and DELETE operations.

DB2LBACWRITEARRAY – Each array component of the user's security label must be equal to the corresponding array component of the data (row or column) security label.

DB2LBACWRITETREE – Each tree component of the user's security label must include at least one of the corresponding elements in the tree component of the data (row or column) security label (or the ancestor of one such element).

DB2LBACWRITESET – Each set component of the user's security label must include the corresponding set component of the data (row or column) security label.

In addition to the rules cited above, DB2 offers specific exemptions that can be assigned to users to bypass one or more of the read and write rules summarized above.

Inappropriate reuse of data in allocated resources is prevented by allowing data to be read only after it has been written thereby allocating resources. This prevents the leakage of information from an authorized user to one that does not have the proper access privileges.

In order to protect against inadvertent database operations, a user can rollback the statements (i.e., operations that can be expressed as SQL) they have issued as long as they haven't been committed. The user can only roll back all of the statements that have occurred since the last time they were committed.

The Access Control security function satisfies the following security requirements:

FDP_ACC.1 Subset access control – DB2 fulfills this requirement by associating privileges with all operations applicable to each identified DB2 object and requiring that a user have the privilege or authority, directly or indirectly through group or database role membership, when attempting to perform the corresponding operation.

FDP_ACF.1 Security attribute based access control – DB2 fulfills this requirement by associating privileges with all operations applicable to each identified DB2 object and requiring that a user have the privilege or authority when attempting to perform the corresponding operation.

FDP_IFC.1 Subset information flow control – DB2 fulfills this requirement by allowing tables to be assigned LBAC policies that will control subsequent read and write operations.

FDP_IFF.2 Hierarchical security attributes – DB2 fulfills this requirements by enforcing the LBAC information flows rules as summarized above.

FDP_RIP.2 Full residual information protection – DB2 fulfills this requirement by ensuring that data can only be read after it has first been written.

FDP_ROL.1 Basic Rollback – DB2 fulfills this requirement by allowing users to rollback any uncommitted statements in the reverse order that they occurred.

6.1.3 Identification & Authentication

If a user attempts to access DB2 without a user ID and password while logged on to the DB2 host operating system (i.e., operational environment), DB2 will derive an authorization name ("authid") from the user ID of the user's host process. This is based on the assumption that the host has already identified and authenticated the user. An authid is the name DB2 uses to identify users, groups, and database roles.

When a user attempts to access DB2 remotely (i.e., while not logged onto the DB2 host operating system), they must provide a user identity and password. The user identity and password are provided to the DB2 host operating system, configured LDAP server, or configured KDC server (using Kerberos) – depending on the TOE configuration - for authentication. If the configured authentication server determines that the user identity exists and the password is valid[8], it will respond to DB2 with the authenticated user identity and any applicable group memberships. Otherwise, it will return a failed result that will cause DB2 to reject the request.

Once authenticated:

- The user must be identified to DB2 using a SQL authorization name or authorization ID (authid). This name can be the same as the user ID, or a mapped value. For example, DB2 authids are usually derived by transforming the user ID to all uppercase letters. If the resulting authid fails to follow any DB2 naming conventions (e.g., allowed characters or size), then DB2 will reject the access request. Note that the DB2 name requirements vary depending on the underlying authentication server (operating system, LDAP, or Kerberos). This is necessary to make sure that the names are uniquely mapped. These restrictions are fully explained in the administrator guidance.
- A list of groups to which the user belongs is obtained. Groups are DB2 host operating system constructs that must also map to DB2 authorization names. This mapping is done in a method similar to that used for user IDs. However, any groups that fail to follow DB2 naming conventions will be ignored, but the access attempt will not be rejected on this basis.
- Once the user identity and groups are established, DB2 will examine its role configuration and assign the roles associated with the user identity, all groups, and other roles (in a potentially hierarchical manner) the user is a member of to the resulting user session.

If an authid is resolved for the user, it becomes the initial user identity used, for example, to enforce the DAC Policy (i.e., the initial "session authorization id"). The authid for the user is used to determine the security label and any LBAC-related exemptions for the session. Authids derived from group and database role memberships are also used for access control. Once the identification and authentication process successfully yields one or more authids, DB2 instantiates a session with those authids for DB2 to allow mediated DB2 operations. Furthermore, DB2 associates specific authorities with the authids available to the user – this is also known as "Authorization."

Trusted users defined by trusted context objects are handled a little differently. Initially they connect essentially like any other client. They must be identified an authenticated as indicated above, but they are additionally identified by their associated trusted context object. A trusted context object can be associated with a user only if they satisfy the conditions defined within the trusted context object (i.e., authorization name, encryption attribute, and IP address).

After connecting, trusted users can change the authorization name of any of their connections to any other authorization name that is authorized within the trusted context object associated with that trusted user. If no such list is defined, then the trusted user cannot change authorization names. In either case, the TOE can be configured to require a password in which case the password must allow the TOE to ensure the new authorization name is successfully authenticated (using the configured user authentication mechanism) or the change will fail. If the authorization name is not valid, the change will simply fail and the connection will remain unchanged.

[8] Note that in addition to the password being correct, it must also not be expired. Additionally, some authentication servers (e.g., host operating systems) have additional identification and authentication conditions that can cause additional authentication failures (account locked, time of day restrictions, workstation restrictions, etc.).

Once the authorization name is changed on a connection, the connection is treated as though the user just authenticated – see *Once authenticated*, above. As such, the connection will subsequently be associated with the new authorization name and the security attributes (including authorizations, roles, security labels, and exemptions) associated with that authorization name.

In addition, the trusted context object can define a default role and also define specific roles for any of its defined authorization names. During initial trusted context connection and also during subsequent authorization name changes within a trusted context, if the applicable authorization name has a database role defined within the trusted context object that database role will be assigned to the session. Otherwise, if there is a default database role defined within the trusted context object that database role will be assigned to the session. If neither are defined, then no additional database role will be assigned to the session.

Note that DB2 includes a set session authorization which allows users with that authorization to change their session identity. This authorization can be granted only by a Security Administrator who is instructed not to grant this authority to any non-administrative users and administrative users are expected not to use this function in the evaluated configuration.

The Identification & Authentication security function satisfies the following security requirements:

> *FIA_ATD.1 User attribute definition* – DB2 fulfills this requirement by maintaining a correspondence between authids, authorities, security labels, and exemptions associated with users.
>
> *FIA_UAU_EXP.2 User authentication before any action* – DB2 fulfills this requirement by rejecting access to DB2 resources when the user cannot be authenticated using support from its operational environment. Note that requirements are also instantiated in its operational environment such that it similarly protects itself and ensures appropriate strength of this mechanism.
>
> *FIA_UID.2 User identification before any action* – DB2 fulfills this requirement by rejecting access to DB2 resources when the user cannot be identified.
>
> *FIA_USB.1 User-subject binding* – DB2 fulfills this requirement by associating authids as well as authorities, security labels, and exemptions with user sessions. While in most cases users cannot change their identities, trusted contexts allow associated trusted users to change their connection identity to other users in accordance with their trusted context definition.

6.1.4 Security Management

All access control to objects subject to the Discretionary Access Control (DAC) security policy as well as to TSF data and functions are controlled using authorities and privileges. DB2 defines a number of authorities and privileges, which allow authorized users and administrators to perform specific functions or access specific resources. These authorities and privileges are assigned to objects using DB2 tables and configuration files (i.e., access control lists) that are similarly controlled with authorities and privileges. Members of the “user” role are most directly subject to the DAC policy and prevented from modifying the behavior of the TSF.

Privileges enable users to create, modify, or access database resources. Authority levels provide a method of grouping privileges and higher-level database manager maintenance and utility operations. Together, these act to control access to the database manager and its database objects. Users can access (including attribute modification and revocation) only those objects for which they have the appropriate authorization, that is, the required privilege or authority. Note that every object that can be created in DB2 will be assigned a default set of security attributes; however, the specific value of the attributes may vary from object to object. However, the default security attributes are predefined by DB2 and cannot be modified by any user. Note that of all of the protected DB2 objects, only databases assign default privileges to users other than the creator (e.g., public). Even so, other users have only limited access to the database. Once another user connects to the database, they can create tables, packages, and schemas within it. Given that databases don’t actually store information, but rather store other objects that contain information and those do not grant any access by default, the overall default access is considered ’restrictive’.

LBAC security attributes (i.e., security labels) for users can only be assigned and modified by a user with SECADM authority (i.e., a security administrator) and only when the TOE has been properly configured with LBAC. Note that generally LBAC policies (i.e., LBAC protection applied a table) are not assigned to objects by default, but rather must be explicitly assigned to each applicable object (i.e., table). The exception to this is when a new row is added to a labeled table. In that case, the new role will be assigned the security label of the user adding the row unless that subject explicitly provides a label in accordance with the LBAC policy rules. DB2 provides the administrator functions to create and drop LBAC policies and labels and to grant and revoke security labels and exemptions to users and to create security policies and security labels (and security label components) that can be used to control access (i.e., protect) to either rows or columns within the table. Once a table is protected, access and modification to specific rows or columns is based on the user security label, security labels within the table, and the LBAC access rules.

DB2 allows users with SECADM authority to create and drop database roles. Database roles are similar to groups, except that they are defined and managed within DB2 rather than its underlying operating system. Users with SECADM authority can grant and revoke roles without restriction. Roles can be granted with an admin option that allows the granted user to also grant and revoke that role, but *without* the admin option.

For privileges associated with a user identity or with a group or database role membership, the privilege may be revoked from the user. In some cases, specifically when the privilege is granted via group membership, the change in privilege may not be immediately effective. To make the change effective immediately, any existing database connections associated with the user may be "forced" (disconnected) by a system administrator. Changes to privileges granted directly to a user or via a database role membership are effective immediately after the change. Changes to access rights associated with an object are not effective until the next access check that would normally be required.[9]

As described in Section 6.1.1, Security Audit, DB2 provides system administrators with the functions necessary to start and stop the audit security function, as well as the tools necessary to create and delete audit files and to configure the audit security function to control specifically which auditable events will be audited and also to export and subsequently review the collected audit records. Note that system administrator, security administrators, and user granted execute to the audit review and extract functions can access audit data. Furthermore, DB2 allows security administrators to define audit policies to fine tune the audit selection function.

In addition to system administrators, security administrators, and other users, DB2 recognizes trusted users defined by trusted context objects. Trusted users have the unique ability to change their identity outside the scope of TOE authentication. This is addressed in the user-subject binding rules, but otherwise such users a treated like other users albeit with a special privilege.

The following is a summary of the authorities defined within DB2:

SYSADM (System Administrator)

> A user with SYSADM authority is not considered a database administrator and has no inherent privilege in the database. Users with system administrator authority have sufficient authority to run most DB2 utility programs, issue database manager commands, maintain database partition groups, table spaces, and bufferpools. Any SYSADM user requiring data access must be granted explicit privileges. By default, the database creator, who must hold SYSADM or SYSCTRL authority, will be granted DBADM, DATAACCESS, ACCESSCTRL, and SECADM during database creation and those authorities can be subsequently revoked by an authorized user. This role is assigned to a user via membership in the operating system SYSADM_GROUP.

SECADM (Security Administrator)

> The SECADM authority is required to perform database security administration and essentially has full control of database security. The security administrator may create, drop, and alter (where applicable) roles, audit, trusted contexts, security labels, security label components, and security policies. In addition, the

[9] Note that when a user creates a package with static data manipulation statements, authorization checks are made to ensure the user is allowed to perform all of the included statements. Subsequently, when a user executes the package, the only authorization check is whether they are authorized to execute the package – the included statements execute based on the authorities of the user that created the package. However, if any revoke operations affect any statements in any defined package, the affected packages are invalidated and are no longer available for execution. Packages already being executed at that time would complete, but would then no longer be accessible.

authority allows a user to grant and revoke all database authorities/privileges. The SECADM authority does not allow a security administrator to access data, however it is possible for the user to obtain DATAACCESS indirectly by granting the authority to one of their groups or roles.

DBADM (Database Administrator)

The DBADM authority is intended to allow management of a database, but the authority can be limited depending on whether ACCESSCTRL or DATAACCESS are also granted. DBADM authority no longer inherently grants additional database level authorities to the applicable authorization id. However, the following authorities will are available to the database administrator as long as the user holds DBADM authority, but will be lost if the authority is revoked

BINDADD
CONNECT
CREATETAB
CREATE_EXTERNAL_ROUTINE
CREATE_NOT_FENCED_ROUTINE
IMPLICIT_SCHEMA
QUIESCE_CONNECT
LOAD

Access Control Authority (ACCESSCTRL)

The ACCESSCTRL authority provides the holder with the ability to issue the following grant and revoke statements on the database, global variables, indices, packages, routines, schemas, sequences, servers, tables, views, nicknames, table spaces, workloads, and XSR objects in the context of the database where they have been assigned this authority.

Note that in DB2 product versions prior to v9.7 ACCESSCTRL authority was held implicitly by all database administrators. In order to preserve existing DB2 behaviour, the GRANT DBADM syntax provides two new options: WITH ACCESSCTRL and WITHOUT ACCCESSCTRL. When only GRANT DBADM is specified, this is considered equivalent to GRANT DBADM WITH ACCESSCTRL. Users who wish to grant database administrator without ACCESSCTRL authority must explcitly state GRANT DBADM WITHOUT ACCESSCTRL in the SQL syntax.

Data Access Authority (DATAACCESS)

The DATAACCESS authority allows the database administrator to be restricted from accessing data in the database tables. Users with this authority can issue the database load statement; issue the select, insert, updated, and delete statmenets on tables, views, and nicknames; and, execute packages and routines (except further restricted audit routine).

Note that, like ACCESSCTRL, DATAACCESS authority was previosuly held implicitly by all database administrators. In order to preserve existing DB2 behaviour, the GRANT DBADM syntax provides two new options: WITH DATAACCESS and WITHOUT DATAACCESS. When only GRANT DBADM is specified, this is considered equivalent to GRANT DBADM WITH DATAACCESS. Users who wish to grant database administrator without DATAACCESS authority must explcitly state GRANT DBADM WITHOUT DATAACCESS in the SQL syntax.

System Control (SYSCTRL)

The SYSCTRL authority level provides control over operations that affect system resources. For example, a user with SYSCTRL authority can create, update, start, stop, or drop a database. This user can also start or stop an instance, but cannot access table data. Users with SYSCTRL authority also have SYSMON authority. This role is assigned to a user via membership in the operating system SYSCTRL_GROUP.

System Monitor (SYSMON)

SYSMON provides the authority required to use the database system monitor. It operates at the instance level. This role is explicitly assigned to a user via membership in the operating system SYSMON_GROUP.

System Maintenance (SYSMAINT)

The SYSMAINT authority level provides the authority required to perform maintenance operations on all databases associated with an instance. A user with SYSMAINT authority can update the database configuration, backup a database or table space, restore an existing database, and monitor a database. Like SYSCTRL, SYSMAINT does not provide access to table data. Users with SYSMAINT authority also have SYSMON authority. This role is assigned to a user via membership in the operating system SYSMAINT_GROUP.

The following additional authorities are considered not security relevant in the context of this Security Target:

SQLADM (SQL Administrator)

The SQLADM authority offers the ability to perform the following actions:

CREATE EVENT MONITOR
DROP EVENT MONITOR
SET EVENT MONITOR STATE
FLUSH EVENT MONITOR
EXPLAIN
FLUSH OPTIMIZATION PROFILE CACHE
FLUSH PACKAGE CACHE
PREPARE
REORG INDEXES/TABLE
RUNSTATS

WLMADM (Workload Manager Administrator)

The WLMADM authority offers the ability to issue grant and revoke statements for workload[10] privileges and to perform the following actions:

ALTER: service class, threshold, work action set, work class set, workload
COMMENT: service class,threshold,work action set, work class set, workload
CREATE: service class, threshold, work action set, work class set, workload
DROP: service class, threshold, work action set, work class set, workload
GRANT: workload privileges
REVOKE: workload privileges
EXECUTE: privilege on workload management routines

The Security Management security function satisfies the following security requirements:

FMT_MOF.1 Management of security functions behaviour – DB2 fulfills this requirement by restricting the abilities to create and drop LBAC security labels, label components and policies as well as to grant and revoke security policies and LBAC exemptions to the security administrator.

FMT_MSA.1a Management of Security Attributes – DB2 fulfills this requirement by allowing only users with the appropriate privilege to modify the Discretionary Access Control security attributes of any DB2 object, including database role assignments.

FMT_MSA.1b Management of Security Attributes – DB2 fulfills this requirement by allowing only LBAC security attributes of rows and columns within DB2 tables to be assigned according to the LBAC access rules.

[10] Note that workloads are not considered to be security relevant objects for the purpose of this evaluation.

FMT_MSA.3a Static Attribute Initialization – DB2 fulfills this requirement by ensuring that objects are assigned restrictive default security attributes when created.

FMT_MSA.3b Static Attribute Initialization – DB2 fulfills this requirement by not assigning LBAC security policies to (i.e., 'protecting') tables by default; the LBAC policy on a given table must be explicitly assigned. However, once a table is protected rows added to the label will be labeled by default with the user's security label unless they explicitly provide a security label in accordance with the LBAC policy rules.

FMT_MTD.1a Management of TSF data – DB2 fulfills this requirement by allowing only system administrators to start the audit service (thereby creating an audit trail) or delete audit records.

FMT_MTD.1b Management of TSF data – DB2 fulfills this requirement by allowing only system and security administrators to define and access the audit selection criteria (at a broad level for system administrators and via audit policies for security administrators).

FMT_MTD.1c Management of TSF data – DB2 fulfills this requirement by allowing only security administrators to create database roles.

FMT_REV.1 Revocation – DB2 fulfills this requirement by allowing only users with the appropriate privilege or authority to modify (including revoke) the security attributes of any DB2 object.

FMT_SMF.1 Specification of Management Functions – DB2 fulfills this requirement by providing functions that allow an authorized administrator to start and stop the audit function; select audited events; create, delete, and review the audit trail; create LBAC policies and labels; grant and revoke LBAC security labels and exemptions; create, drop, grant, and revoke database roles; and grant and revoke DAC access attributes.

FMT_SMR.1 Security Management Roles – DB2 fulfills this requirement by defining system, security, and database administrator and user roles. The user role can perform security management functions as allowed by the DAC and LBAC rules, ACCESSCTRL authority (if held), and available privileges to access audit data. There is also a trusted context role which serves to differentiate trusted users from other TOE clients.

6.1.5 TOE Protection

DB2 is designed to operate within a set of processes provided by the hosting operating system. DB2 does not support the ability to share its processes with non-TOE entities. Note that DB2 supports both 'fenced' and 'unfenced' routines. Fenced routines execute in their own process distinct from that of the DB2 server, while unfenced routines share the process with the DB2 server. Given that such routines are created by users and as such cannot be subject to evaluation, the evaluated configuration does not include any provisions for unfenced routines (i.e., they are not included in the evaluated configuration of the TOE).

Furthermore, DB2 is designed in a manner that ensures that its interfaces do not offer unauthorized users any functions that might be used to corrupt, or otherwise inappropriately access, the TSF. As is the case with many application-only TOEs such as DB2, its protection mechanisms could be bypassed through the underlying environment should the assumptions (e.g., A.Platform) and objectives (e.g., OE.ENFORCEMENT) for its environment not be fulfilled. Note that determination of fulfillment of those assumptions and objectives is not within the scope of the TOE.

DB2 has been designed to implement a number of DB2-specific objects and functions. Each DB2 object and function is available via interfaces provided by DB2, and each interface has been carefully designed to ensure that it only provides appropriate capabilities or access after necessary security checks have been made and approved.

DB2 has been designed to collect current time information from its hosting operating system in a correct and consistent manner. Once it has been collected, DB2 ensures that it is not corrupted as it is being used by the DB2 TSF, thereby ensuring that it remains reliable.

DB2 utilizes the IBM Global Security Kit (GSKit), which offers several cryptographic functions. In particular, when appropriately configured, DB2 can use the functions of GSKit to establish SSL connections initiated by java-based clients. These connections serve to protect all network communication of sensitive user credentials and user data between the client and DB2 server.

As summarized in section 2.2.4.5, DB2 also provides the optional feature to require that user IDs and passwords are encrypted by the associated user clients in order to offer more protection for user authentication data. As indicated in that section, the AES-256 user ID and password protection feature is implemented using the IBM Crypto for C (ICC) version 1.4.5 library linked with the TOE. Note that ICC v1.4.5 has been FIPS certified (certificate number 775), albeit in conjunction with earlier versions of the operating systems currently hosting DB2.

The TOE Protection security function satisfies the following security requirements:

FPT_STM_EXP.1 Reliable Time Stamps – DB2 fulfills this requirement by consistently collecting time information from its operational environment and then by protecting it while it is being used. Note that a similar requirement is levied on the operational environment to ensure that it also has access reliable timestamps.

FTP_ITC.1 Inter-TSF trusted channel – DB2 fulfills this requirement by being able to establish a secure SSL connection between itself and java-based clients at the request of the client.

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

8. Rationale

This section includes the rationale for the functional and assurance requirements specified for the TOE. The rationale is based on specified objectives, threats, assumptions, and policies.

8.1 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the protection profile.

8.1.1 Complete Coverage - Threats

The TOE security objectives have been derived exclusively from statements of organizational security policy, and therefore, there are no explicitly defined threats countered by this profile.

8.1.2 Complete Coverage - Policy

This section provides evidence demonstrating coverage of the Organizational Security Policy by both the IT and Non-IT security objectives. The following table shows this objective to policy mapping, and the table is followed by a discussion of the coverage for each Security Policy.

Table 4 Mapping of Organizational Security Policies to Security Objectives

Organizational Security Policy	Security Objectives
P.AUTHORIZED_USERS	O.AUTHORIZATION
	OE.AUTHORIZATION
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
	OE.CRYPTO
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS
	O.RESIDUAL_INFORMATION
	OE.RESIDUAL_INFORMATION
	O.ROLLBACK
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
	OE.CRYPTO
P.ACCOUNTABILITY	O.AUDITING
	OE.AUDITING
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
	OE.CRYPTO
P.CLASSIFICATION	O.MANDATORY_ACCESS
	O.RESIDUAL_INFORMATION
	O.MANAGE
	O.ENFORCEMENT
	OE.CRYPTO

The following discussion provides detailed evidence of coverage for each statement of organizational security policy:

P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the TOE may access the TOE.

This policy is primarily realized by the O.AUTHORIZATION and OE.AUTHORIZATION objectives. The O.AUTHORIZATION and OE.AUTHORIZATION objectives require that the TOE and its operational environment provide access only to authorized users. The O.MANAGE and OE.MANAGE objectives support this policy by requiring that an authorized administrator is able to manage the functions. The O.ENFORCEMENT and OE.ENFORCEMENT objectives ensure that functions are invoked and operate correctly. The OE.CRYPTO objective helps to protect sensitive data from disclosure when sent between remote clients and the TOE.

P.NEED_TO_KNOW

The TOE must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a "need to know" for that information.

This policy is primarily realized by the O.DISCRETIONARY_ACCESS objective, which allows authorized users to control access to resources based on user identities. The O.RESIDUAL_INFORMATION and OE.RESIDUAL_INFORMATION objectives ensure that information will not be given to users that do not have a need-to-know when resources are reused. The O.MANAGE and OE.MANAGE objectives support this policy by requiring that an authorized administrator is able to manage the functions. The O.ENFORCEMENT and OE.ENFORCEMENT objectives ensure that functions are invoked and operate correctly. The O.ROLLBACK objective ensures that any inadvertent operations performed on protected resources can be undone. The OE.CRYPTO objective helps to protect sensitive data from disclosure when sent between remote clients and the TOE.

P.ACCOUNTABILITY

The users of the TOE shall be accountable for their actions within the TOE.

This policy is primarily realized by the O.AUDITING and OE.AUDITING objectives by requiring that actions can be recorded in an audit trail. The O.MANAGE and OE.MANAGE objectives support this policy by requiring that an authorized administrator is able to manage the functions. The O.ENFORCEMENT and OE.ENFORCEMENT objectives ensure that functions are invoked and operate correctly. The OE.CRYPTO objective helps to protect sensitive data from disclosure when sent between remote clients and the TOE. Note that while user can be held accountability when audit records are recorded, the TOE only must be able to record audit events. This feature is configurable and, at the discretion of the administrator, selectively audits user activity. Furthermore, the TOE could experience a failure, such as disk space exhaustion, that might prevent audit events from being recorded, but under normal circumstances the TOE will record the events selected by the administrator.

P.CLASSIFICATION

The system must be able to limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at.

This policy is implemented by the O.MANDATORY_ACCESS objective. The O.RESIDUAL_INFORMATION objective ensures that information will not given to users which do not have a cleared access, when resources are reused. The O.MANAGE supports this policy by requiring authorized administrator be able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly. The OE.CRYPTO objective helps to protect sensitive data from disclosure when sent between remote clients and the TOE.

8.1.3 Complete Coverage - Environmental Assumptions

This section provides evidence demonstrating coverage of the Non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

Table 5 Mapping of Environmental Assumptions to Non-IT Security Objectives

Environmental Assumptions	Non-IT Security Objectives
A.MANAGE	O.ASSIGN
	O.INSTALL
A.NO_EVIL_ADM	O.ADMIN_GUIDANCE
	O.ADMINISTRATORS
	O.INSTALL
A.LOCATE	O.PHYSICAL
A.PROTECT	
A.CONNECT	
A.COOP	O.COOP
	O.CREDEN
A.PLATFORM	O.PLATFORM
A.CLEARANCE	O.CREDEN

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This is addressed by O.ASSIGN, which ensures that competent individuals are assigned to manage the TOE and the security of its information, and by O.INSTALL, which ensures that the TOE is delivered, installed, managed and operated in a manner that maintains IT security.

A.NO_EVIL_ADM

The administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

This is primarily addressed by O.ADMINISTRATORS, which ensures that Administrators of the TOE and its operational must not be careless, willfully negligent or hostile, and must follow the instructions provided in the administrator guidance documentation. The O.ADMIN_GUIDANCE objective ensures that administrators receive guidance documentation enabling them to install, manage, and operate the TOE securely. This assumption is also addressed by O.INSTALL, which ensures that the TOE is delivered, installed, managed and operated in a manner that maintains IT security.

A.LOCATE

The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

This is addressed by O.PHYSICAL which addresses those parts of the TOE which are critical to security policy are protected from physical attack.

A.PROTECT

The hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

This is addressed by O.PHYSICAL which addresses those parts of the TOE which are critical to security policy are protected from physical attack.

A.CONNECT

All connections to peripheral devices reside within the controlled access facilities. The TOE only addresses security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths, including networks, to access points and between TOE instances are assumed to be adequately protected.

This is addressed by O.PHYSICAL which ensures that those parts of the physical TOE and its associated operational environment critical to security policy are protected from attack that might compromise IT security objectives.

A.COOP

Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

This is addressed by O.COOP, which ensures that authorized users possess the appropriate authorization to access at least some of the information managed by the TOE and act in a cooperative manner in a benign environment. This is also addressed by O.CREDEN that states that those responsible for the TOE must ensure that all access credentials such as passwords or other authentication information are protected by the users in a manner that maintains IT security objectives.

A.PLATFORM

The Environment underlying the TOE is assumed to fulfill the objectives for the TOE-supporting components in the operational environment described in this ST.

This is addressed by O.PLATFORM that basically reiterates the assumption to expect the Environment to provide a suitable and effective environment for the operation of the TOE.

A.CLEARANCE

Procedures exist for granting users authorization for access to specific security levels. It is further assumed the TOE administrators will be cleared to the highest security level processed by the TOE.

This is addressed by O.CREDEN that states that credentials such as clearances, perhaps represented by security labels, must be associated with user appropriately.

8.2 Security Requirements Rationale

This section provides evidence supporting the combined internal consistency and completeness of the requirements in this Security Target.

8.2.1 Internal Consistency of Requirements

This section describes the mutual support and internal consistency of the components selected for this Security Target. These properties are discussed for both functional and assurance components.

The functional components were selected from pre-defined CC components. Assignment, selection, and refinement operations were carried out among components using consistent computer security terminology. This helps to avoid the ambiguity associated with interpretations of meanings of terms between related components. Multiple instantiation of components was used to clearly state the required functionality that must exist in the TOE.

Each security functional requirement in the ST was selected to avoid conflicts with other security functional requirements in the ST.

The IT security functional requirements form a mutually supportive whole. Table 8 in Section 8.2.2 maps the functional components to security objectives. Table 9 in Section 8.4 demonstrates that the TOE security functional requirement dependencies have been satisfied.

Additionally, Section 5 of the ST contains several security requirements that support other requirements, as detailed in the following table.

Table 6 Security Requirements Supporting Other Requirements

Security functional requirement	Effect
FAU_GEN.1	Detect attempts to bypass or tamper with other security functional requirements
FAU_GEN.2	
FAU_SAR.1	
FAU_STG.4	
ADV_ARC.1	Prevent other security functional requirements from being bypassed
ADV_ARC.1 FAU_STG.1	Prevent other security functional requirements from being tampered with

8.2.2 Complete Coverage - Objectives

This section demonstrates that the functional components selected for this Security Target provide complete coverage of the defined IT security objectives. The mapping of components to IT security objectives is depicted in the following table.

Table 7 Mapping of Security Objectives to Functional Components

Security Objective	Functional Component
O.AUTHORIZATION	FIA_ATD.1
	FIA_UAU_EXP.2
	FIA_UID.2
O.DISCRETIONARY_ACCESS	FDP_ACC.1
	FDP_ACF.1
	FIA_ATD.1
	FIA_USB.1
	FMT_MSA.1a
	FMT_MSA.3a
	FMT_MTD.1c
	FMT_REV.1
O.MANDATORY_ACCESS	FDP_IFC.1
	FDP_IFF.2
	FIA_ATD.1
	FIA_USB.1
	FMT_MOF.1
	FMT_MSA.1b
	FMT_MSA.3b
O.AUDITING	FAU_GEN.1
	FAU_GEN.2
	FAU_SAR.1
	FAU_SAR.2

Security Objective	Functional Component
	FAU_SAR.3
	FAU_SEL.1
	FAU_STG.3
	FAU_STG.4
	FIA_USB.1
	FMT_MTD.1a
	FMT_MTD.1b
	FMT_SMF.1
	FPT_STM_EXP.1
O.RESIDUAL_INFORMATION	FDP_RIP.2
O.ROLLBACK	FDP_ROL.1
O.MANAGE	FAU_SAR.1
	FAU_SAR.3
	FAU_SEL.1
	FAU_STG.3
	FAU_STG.4
	FMT_SMF.1
	FMT_SMR.1
O.ENFORCEMENT	FTP_ITC.1 ADV_ARC.1

The following discussion provides detailed evidence of coverage for each security objective:

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

Users must be identified [FIA_UID.2], authenticated [FIA_UAU_EXP.2], and associated with available authorities and privileges [FIA_ATD.1] before they can access the TOE and the resources it protects.

O.DISCRETIONARY_ACCESS

The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which users may access which resources.

Discretionary access control must have a defined scope of control [FDP_ACC.1]. The rules of the DAC policy must be defined [FDP_ACF.1]. The security attributes of objects used to enforce the DAC policy must be defined [FDP_ACF.1]. Authorized users must be able to control who has access to objects [FMT_MSA.1a and FMT_MTD.1c] and be able to revoke that access [FMT_REV.1]. Default protection must be available from an object's creation [FMT_MSA.3a].

O.MANDATORY_ACCESS

The TSF must be able to control access to resources based upon the sensitivity and categories of the information being accessed and the clearance of the subject attempting to access that information.

Mandatory access control attributes and rules must be definable [FDP_IFF.2] and must have a definable scope of control [FDP_IFC.1]. Finally, if the MAC policy is to be enforced, it is required that it can be enabled and that attributes be associated with each object [FMT_MOF.1, FMT_MSA.1b, FMT_MSA.3b], and that the binding between processes and the attributes of the user on whose behalf they operate be correct and unforgable [FIA_ATD.1, FIA_USB.1].

O.AUDITING

The TSF must be able to record the security relevant actions of users of the TOE. The TSF must be able to present this information only to authorized administrators.

Security-relevant actions must be defined, auditable [FAU_GEN.1], and capable of being associated with individual users [FAU_GEN.2, FIA_USB.1]. The audit trail must be protected so that only authorized users may access it [FAU_SAR.2]. The TSF must provide the capability to audit specific types of actions [FAU_SEL.1] and the actions of individual users [FAU_SAR.3, FIA_USB.1]. The audit facility must have some defined behavior if the audit trail becomes full [FAU_STG.4]. The time stamp associated must be reliable [FPT_STM_EXP.1a]. An authorized administrator must be able to review [FAU_SAR.1], manage [FAU_STG.3, FMT_SMF.1], and protect [FMT_MTD.1a, FMT_MTD.1b] the audit trail.

O.RESIDUAL_INFORMATION

The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

Residual information associated with defined objects in the TOE must be inaccessible during reuse of the object containing the residual information [FDP_RIP.2].

O.ROLLBACK

The TSF must ensure that operations performed on information contained in a protected resource can be undone before the results have been committed.

The TOE must provide a mechanism to undo any operation that can be expressed as SQL performed on a protected resources so long as it has not yet been committed [FDP_ROL.1].

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

The TSF must provide for an authorized administrator to manage the TOE [FMT_SMR.1]. The administrator must be able to review and manage the audit trail [FAU_SAR.1, FAU_SAR.3, FAU_SEL.1, FAU_STG.3, FAU_STG.4, FMT_SMF.1] along with all other security functions of the TOE [FMT_SMF.1].

O.ENFORCEMENT

The TSF must be designed and implemented in a manner that ensures that the organizational policies are enforced in the target environment.

The TSF must make and enforce the decisions of its security policies [ADV_ARC.1]. The TSF must protect itself from interference that would prevent it from performing its functions [ADV_ARC.1]. The correctness of this objective is further met through the assurance requirements defined in this Security Target. This objective provides global support to other security objectives for the TOE by protecting the parts of the TOE, which implement policies and ensures that policies are enforced. The TSF must be able to protect network communication that might otherwise allow one of its policies to be subverted [FTP_ITC.1].

8.3 Assurance Requirements Rationale

The TOE was developed based on the C2 requirements of the Trusted Computer System Evaluation Criteria (TCSEC). Those requirements have been reproduced in the Controlled Access Protection Profile (CAPP) using Common Criteria conventions. While the CAPP demands only EAL 3, this Security Target claims EAL 4 augmented with ALC_FLR.1. This added assurance is intended to provide consumers more confidence in the security features of the TOE so that the product may be used in a wider variety of environments.

8.4 Requirement Dependency Rationale

The following table shows the security functional and assurance requirement dependencies that exist based on the security functional and assurance requirements (and iterations thereof) included in this Security Target. As indicated in the following table all of the dependencies are satisfied with the exception of those of FAU_STG.1, FIA_UAU_EXP.2, and FPT_STM_EXP.1. The TOE is a software application that is dependent upon its host operating system to provide secure audit storage and a reliable source of time (see OE.ENFORCEMENT) and authenticate users on behalf of the TOE (see OE.AUTHENTICATION).

Note that in the left column TOE security functional requirements are identified normally and assurance requirements are underlined.

Table 8 TOE Security Functional Requirement Dependencies

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM_EXP.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.2
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1 and FMT_MTD.1
FAU_STG.3	FAU_STG.1	Not satisfied, see rationale above.
FAU_STG.4	FAU_STG.1	Not satisfied, see rationale above.
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1 and FMT_MSA.3a
FDP_IFC.1	FDP_IFF.1	FDP_IFF.2
FDP_IFF.2	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1 and FMT_MSA.3b
FDP_RIP.2	none	none
FDP_ROL.1	(FDP_ACF.1 or FDP_IFC.1)	FDP_ACF.1 and FDP_IFC.1
FIA_ATD.1	none	none
FIA_UAU_EXP.2	FIA_UID.1 FIA_UAU.1	FIA_UID.2 Not satisfied, see rationale above.
FIA_UID.2	none	none
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1a	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.1
FMT_MSA.1b	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_IFC.1
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1 b	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1 c	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_REV.1	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_STM_EXP.1	FPT_STM.1	Not satisfied, see rationale above.
FTP_ITC.1	none	none
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	ADV_FSP.4 and ADV_TDS.3
ADV_FSP.4	ADV_TDS.1	ADV_TDS.3

ST Requirement	CC Dependencies	ST Dependencies
ADV_IMP.1	ADV_TDS.3 and ALC_TAT.1	ADV_TDS.3 and ALC_TAT.1
ADV_TDS.3	ADV_FSP.4	ADV_FSP.4
AGD_OPE.1	ADV_FSP.1	ADV_FSP.4
AGD_PRE.1	none	none
ALC_CMC.4	ALC_CMS.1 and ALC_DVS.1 and ALC_LCD.1	ALC_CMS.4 and ALC_DVS.1 and ALC_LCD.1
ALC_CMS.4	none	none
ALC_DEL.1	none	none
ALC_DVS.1	none	none
ALC_FLR.1	none	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1
ATE_COV.2	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.4 and ATE_FUN.1
ATE_DPT.2	ADV_ARC.1 and ADV_TDS.3 and ATE_FUN.1	ADV_ARC.1 and ADV_TDS.3 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.4 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.2 and ATE_FUN.1
AVA_VAN.3	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.3 and ADV_IMP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and ADV_FSP.4 and ADV_TDS.3 and ADV_IMP.1 and AGD_OPE.1 and AGD_PRE.1

8.5 Extended Requirements Rationale

This Security Target contains two extended requirements: FIA_UAU_EXP.2 and FPT_STM_EXP.1. Each of these requirements is based on the CC versions of FIA_UAU.2 and FPT_STM.1, except that these extended versions specifically allow the environment of the TOE to perform some aspect of the requirement which is not allowed in the original requirements. In the case of FIA_UAU_EXP.2, the authentication function while the TOE enforces restrictions on services until its environment confirms the authenticity of applicable users. In the case of FPT_STM_EXP.1, the environment of the TOE provides timestamps that are subsequently collected, protected, and used by the TOE. Note that the functions implied by these requirements are completely fulfilled by a combination of the TOE and its environment and as such should be considered to satisfy any dependencies levied by other requirements on FIA_UAU.2 or FPT_STM.1.

Both FIA_UAU_EXP.2 and FPT_STM_EXP.1 share the same functional requirement class and family as their CC-defined counterparts. They also share the same dependencies as well as being dependent upon instances of their CC counterparts (FIA_UAU.1 and FPT_STM.1, respectively), but are not hierarchical to any CC-defined requirements. They are otherwise completely defined below:

8.5.1 FIA_UAU_EXP.2 User authentication before any action

Hierarchical to: No other components

Dependencies: FIA_UID.1, FIA_UAU.1

Audit:

Minimal:	Unsuccessful use of the authentication mechanism.
Basic:	All use of the authentication mechanism.

Management: There are no management activities foreseen.

FIA_UAU_EXP.2.1 The TSF shall require each user to be successfully authenticated using support from its environment before allowing any other TSF-mediated actions on behalf of that user.

8.5.2 FPT_STM_EXP.1 Reliable time stamps

Hierarchical to: No other components

Dependencies: FPT_STM.1

Audit: None.

Management: There are no management activities foreseen.

FPT_STM_EXP.1.1 The TSF shall be able to provide reliable time stamps based on information provided by its environment for its own use.

8.6 TOE Summary Specification Rationale

The following table describes the association between the TOE Security Functions and the TOE Security Functional Requirements. This table in conjunction with rationale provided in Section 6.1 demonstrates that the TOE Security Functional Requirements are satisfied.

Table 9 Security Function to TOE SFR Mapping

	Security audit	User data protection	Identification and authentication	Security management	Protection of the TSF
FAU_GEN.1	X				
FAU_GEN.2	X				
FAU_SAR.1	X				
FAU_SAR.2	X				
FAU_SAR.3	X				
FAU_SEL.1	X				
FAU_STG.3	X				
FAU_STG.4	X				
FDP_ACC.1		X			
FDP_ACF.1		X			
FDP_IFC.1		X			
FDP_IFF.2		X			
FDP_RIP.2		X			
FDP_ROL.1		X			
FIA_ATD.1			X		
FIA_UAU_EXP.2			X		
FIA_UID.2			X		
FIA_USB.1			X		
FMT_MOF.1				X	
FMT_MSA.1a				X	

	Security audit	User data protection	Identification and authentication	Security management	Protection of the TSF
FMT_MSA.1b				X	
FMT_MSA.3a				X	
FMT_MSA.3b				X	
FMT_MTD.1a				X	
FMT_MTD.1b				X	
FMT_MTD.1c				X	
FMT_REV.1				X	
FMT_SMF.1				X	
FMT_SMR.1				X	
FPT_STM_EXP.1					X
FTP_ITC.1					X