

IBM Corporation

WebSphere Federation Server

v9.1

Security Target

Revision 1.0
July 15, 2007

Prepared for:



IBM
Silicon Valley Lab
555 Bailey Ave.
San Jose, CA 95141-1003

Prepared By:



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

TABLE OF CONTENTS

1. SECURITY TARGET INTRODUCTION	5
1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION	5
1.2 CONFORMANCE CLAIMS	5
1.3 CONVENTIONS, TERMINOLOGY, ACRONYMS	5
1.3.1 Conventions	5
1.3.2 Acronyms	6
1.4 SECURITY TARGET OVERVIEW AND ORGANIZATION	7
2. TOE DESCRIPTION	8
2.1 PRODUCT TYPE	10
2.2 PRODUCT DESCRIPTION	10
2.2.1 DRDA Protocol Handler	10
2.2.2 SQL Processing	11
2.2.2.1 SQL Manager	11
2.2.2.2 SQL Compiler	12
2.2.2.3 SQL Runtime	12
2.2.3 Non-SQL Processing	12
2.2.4 Optional Features	12
2.2.4.1 DPF	12
2.3 PRODUCT FEATURES	12
2.4 SECURITY ENVIRONMENT TOE BOUNDARY	13
2.4.1 Physical Boundaries	13
2.4.1.1 Wrappers	13
2.4.2 Logical Boundaries	15
2.4.2.1 Security Audit	15
2.4.2.2 Access Control	15
2.4.2.3 Identification & Authentication	16
2.4.2.4 Security Management	16
2.4.2.5 TOE Protection	16
3. SECURITY ENVIRONMENT	17
3.1 SECURE USAGE ASSUMPTIONS	17
3.1.1 Personnel Assumptions	17
3.1.2 Physical Assumptions	17
3.1.3 Connectivity Assumptions	17
3.2 THREATS	18
3.3 ORGANIZATION SECURITY POLICIES	18
4. SECURITY OBJECTIVES	19
4.1 SECURITY OBJECTIVES FOR THE TOE	19
4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT	19
4.2.1 Non-IT security objectives for the environment	19
4.2.2 IT security objectives for the environment	20
5. IT SECURITY REQUIREMENTS	21
5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS	21
5.1.1 Security Audit (FAU)	22
5.1.2 User Data Protection (FDP)	23
5.1.3 Identification and authentication (FIA)	25
5.1.4 Security management (FMT)	26
5.1.5 Protection of the TSF (FPT)	27

5.2	SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT	27
5.2.1	<i>Security Audit (FAU)</i>	28
5.2.2	<i>User Data Protection (FDP)</i>	29
5.2.3	<i>Identification and authentication (FIA)</i>	29
5.2.4	<i>Security management (FMT)</i>	29
5.2.5	<i>Protection of the TSF (FPT)</i>	30
5.3	TOE SECURITY ASSURANCE REQUIREMENTS	30
5.3.1	<i>Configuration Management (ACM)</i>	31
5.3.2	<i>Delivery and Operation (ADO)</i>	33
5.3.3	<i>Development (ADV)</i>	33
5.3.4	<i>Guidance Documents (AGD)</i>	36
5.3.5	<i>Life Cycle Support (ALC)</i>	37
5.3.6	<i>Security Testing (ATE)</i>	38
5.3.7	<i>Vulnerability Assessment (AVA)</i>	39
6.	TOE SUMMARY SPECIFICATION	42
6.1	TOE SECURITY FUNCTIONS	42
6.1.1	<i>Security Audit</i>	42
6.1.2	<i>Access Control</i>	44
6.1.3	<i>Identification & Authentication</i>	46
6.1.4	<i>Security Management</i>	47
6.1.5	<i>TOE Protection</i>	49
6.2	TOE SECURITY ASSURANCE MEASURES	49
6.2.1	<i>Process Assurance</i>	50
6.2.1.1	<i>Configuration Management</i>	50
6.2.1.2	<i>Life cycle support</i>	50
6.2.2	<i>Delivery and Guidance</i>	50
6.2.3	<i>Development</i>	51
6.2.4	<i>Tests</i>	51
6.2.5	<i>Vulnerability Assessment</i>	52
6.2.5.1	<i>Evaluation of Misuse</i>	52
6.2.5.2	<i>Strength of TOE Security Functions</i>	52
6.2.5.3	<i>Vulnerability Analysis</i>	52
7.	PROTECTION PROFILE CLAIMS	53
8.	RATIONALE	54
8.1	SECURITY OBJECTIVES RATIONALE	54
8.1.1	<i>Complete Coverage - Threats</i>	54
8.1.2	<i>Complete Coverage - Policy</i>	54
8.1.3	<i>Complete Coverage - Environmental Assumptions</i>	55
8.2	SECURITY REQUIREMENTS RATIONALE	57
8.2.1	<i>Internal Consistency of Requirements</i>	57
8.2.2	<i>Complete Coverage - Objectives</i>	58
8.3	ASSURANCE REQUIREMENTS RATIONALE	61
8.4	REQUIREMENT DEPENDENCY RATIONALE	61
8.5	EXPLICITLY STATED REQUIREMENTS RATIONALE	63
8.6	TOE SUMMARY SPECIFICATION RATIONALE	64
8.7	STRENGTH OF FUNCTION (SOF) RATIONALE	64

LIST OF TABLES

Table 1	TOE Functional Security Requirements	21
Table 2	Auditable Events for the TOE	22
Table 3	IT Environment Functional Security Requirements	27

Table 4 Auditable Events for the IT Environment.....	28
Table 5 Assurance Requirements (EAL 4 augmented).....	30
Table 6 Mapping of Organizational Security Policies to Security Objectives.....	54
Table 7 Mapping of Environmental Assumptions to Non-IT Security Objectives.....	55
Table 8 Mapping of Security Objectives to Functional Components.....	58
Table 9 TOE Security Functional Requirement Dependencies	62

LIST OF FIGURES

Figure 1 TOE Security Environment.....	8
Figure 2 DB2 and WebSphere FS Components	9
Figure 3 WebSphere FS Trusted Mode	14
Figure 4 WebSphere FS Fenced Mode.....	15

1. Security Target Introduction

This Security Target (ST) describes the IT security requirements for the IBM WebSphere Federation Server v9.1 (with Fix Pack 1C), herein referred to as WebSphere FS. WebSphere FS is a middleware product based on the DB2 Enterprise Server Edition Version 9.1.1 (also known as V9.1 Fix Pack 1) for Linux, Unix, and Windows; herein collectively referred to as DB2. DB2 is a Relational Database Management System (RDBMS) developed by IBM, Silicon Valley Lab, 555 Bailey Ave., San Jose, CA 95141-1003.

The supporting DB2 product 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 WebSphere FS is a middleware product that extends the DB2 RDBMS and not a complete Operating System, some of the requirements have been assigned to the IT environment (i.e., underlying operating system) and conformance cannot, therefore, be claimed in this ST.

1.1 Security Target, TOE and CC Identification

ST Title – IBM Corporation WebSphere Federation Server v9.1 Security Target

ST Version – Revision 1.0

ST Date – July 15, 2007

TOE Identification – IBM DB2 Enterprise Server 9.1.1 with IBM WebSphere Federation Server, v9.1 and Fix Pack 1C

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005.

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 2.3, August 2005.
 - Part 2 extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.3, August 2005.
 - Part 3 conformant
 - 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 v2.3.

- 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 ...”).
- Explicitly stated requirements (i.e., those not taken from the CC) are identified by ‘(**explicitly stated requirement**)’ which will appear after the requirement label. Note that the explicitly stated 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 WebSphere FS Target of Evaluation (TOE) is a Database System offering a wide range of database related services. This ST describes the WebSphere FS TOE, intended environments, security objectives, security requirements (for the TOE and IT environment), security functions, and all necessary rationale. This information is organized into the following additional sections:

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

2. TOE Description

WebSphere FS is middleware product provided by IBM. As middleware (with an embedded RDBMS), WebSphere FS supports the Structured Query Language (SQL) interface from a client that is connected to the server. From the client, commands can be entered interactively or through an executing program to the server to create databases, database tables, and to store and retrieve information from tables either in the embedded DB2 instance or in other associated data sources. WebSphere FS can be installed on a number of possible operating environments.

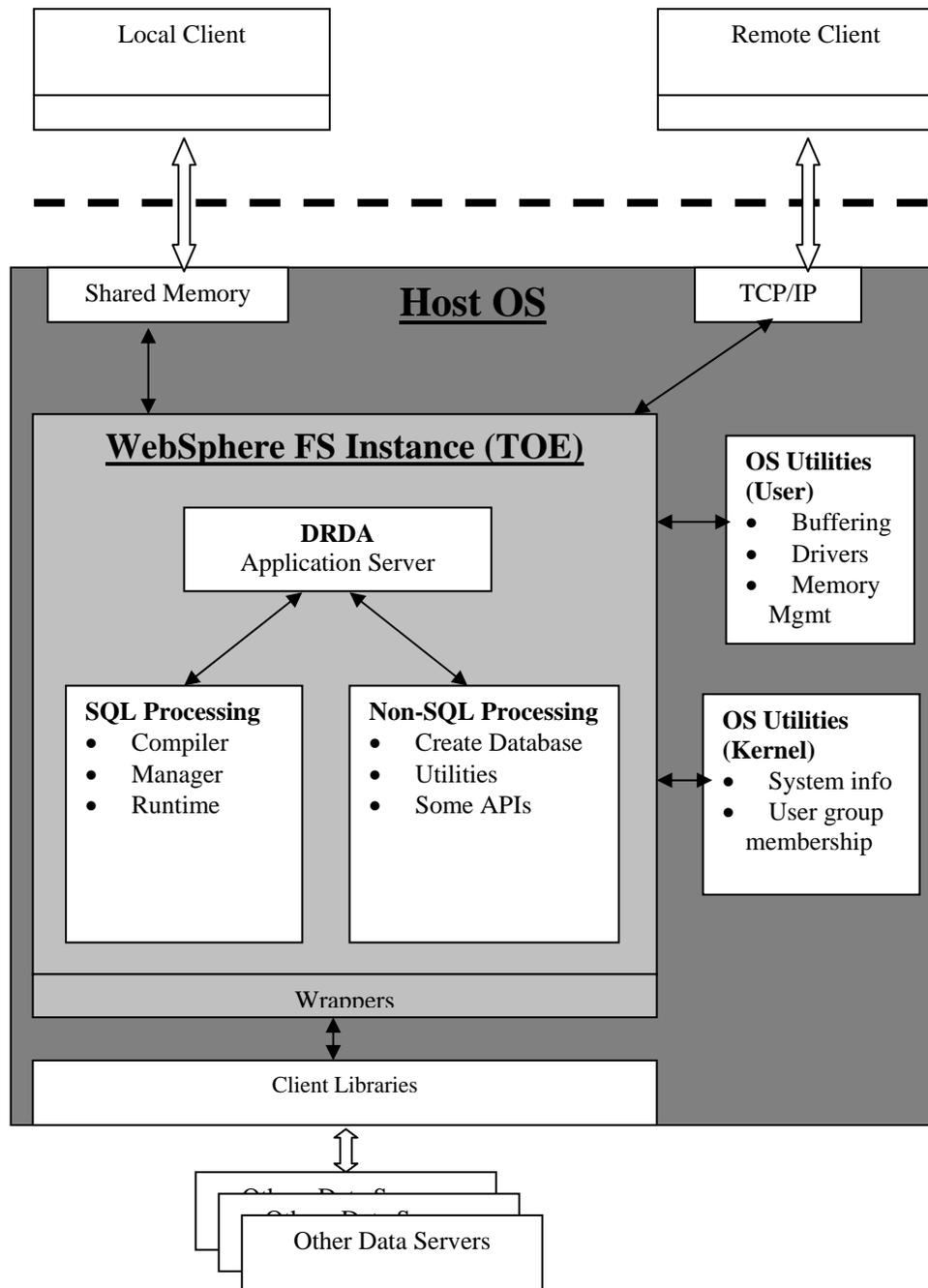


Figure 1 TOE Security Environment

The following figure provides an abstract view of how DB2 and WebSphere FS components are organized to offer a complete WebSphere FS product. The components identified in yellow are shipped with DB2. The components in green are shipped as part of WebSphere FS. The components in red are provided by the data source developers.



Figure 2 DB2 and WebSphere FS Components

The TOE for the WebSphere FS configuration includes all components within the lightly shaded box of Figure 1 TOE Security Environment entitled “WebSphere FS Instance”. 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, the ‘WebSphere FS Instance’ may actually be distributed across multiple WebSphere FS partitions acting together to form that logical view.

Note that WebSphere FS is based primarily on DB2 v9.1. There are some internal modifications, but the most obvious differences are a series of wrappers that allow WebSphere FS to interact with other data sources (other database products as well as non-database data sources such as flat files) using corresponding client libraries.

All other functions are allocated to the IT Environment, which in this case is the Host Operating System (OS), inside the darkly shaded box of Figure 1 TOE Security Environment. The WebSphere FS software is tightly linked to the OS and in some cases, security functions are allocated to the OS, as appropriate. For the purposes of this ST, the OS is AIX 5.3 and Linux RHEL 4. Note that while the product has been designed to operate on other OSs, such as other versions of Linux, Microsoft Windows Servers, and Sun Solaris, the evaluated configuration has been limited for practicality.

2.1 Product Type

WebSphere FS is middleware providing interoperability among a number of configurable data sources. It includes an embedded DB2 multi-user RDBMS that operates in the context of a hosting operating system and allows authorized users to create and manage databases in both the embedded DB2 instance and the other configured data sources.

WebSphere FS operates as a set of applications (e.g., servers) in an IT Environment consisting of all software residing on the host platform(s) but not part of the WebSphere FS TOE. For the purposes of this discussion it is referred to as the Host OS (refer to Figure 1 TOE Security Environment). The IT Environment 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 WebSphere FS. These functions are depicted as individual blocks within the WebSphere FS Instance (TOE) box in Figure 1 TOE Security Environment. WebSphere FS is implemented using the concept of a WebSphere FS instance where an instance is a complete environment dictating what can be done to data and managing the system resources assigned to it. A WebSphere FS instance has one or more databases under its control that cannot be directly accessed by any other instance.

WebSphere FS 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 WebSphere FS 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 system administrator authority.

For the most part the security functions and interfaces of WebSphere FS are the same as those of embedded DB2 v9.1. However, the wrappers introduce some new interfaces that are driven by the TOE. The first interface is the interface to the wrapper itself. This is a common interface regardless of the wrapper and is well documented to facilitate the development of other pluggable wrappers. The second interface is between a given wrapper and the corresponding client (e.g., Oracle client library). This interface is specific and unique for each given wrapper. Note that the evaluated configuration includes a specific set of wrappers and as a result the first interface characterized above is an internal subsystem or module interface. The second interface is largely not security relevant since it is activated by the TOE and not from external sources and the operation of the external clients could not practically be claimed and is therefore outside the context of evaluation.

The following subsections serve to summarize the security aspects of the embedded DB2 instance and while not implemented in WebSphere FS proper, they are part of the composed TOE.

2.2.1 DRDA Protocol Handler

The DRDA Application Server (AS) module within WebSphere FS allows for WebSphere FS 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 WebSphere FS DRDA AS module architecture provides

support for one or more DRDA Application Requestors (DRDA AR), commonly referred to as clients, to access a specific WebSphere FS instance or WebSphere FS database and issue SQL and non-SQL requests against that object. Upon initiation of communication between a client and the WebSphere FS 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 WebSphere FS for processing. These requests are of two general types: SQL requests, which are handled by the WebSphere FS SQL Processing module, and non-SQL requests, which are handled by the WebSphere FS 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 WebSphere FS SQL Processing module is responsible for the analysis and execution of client requests related to the processing of Structured Query Language (SQL) statements. WebSphere FS 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. Such statements would include the LOCK TABLE and SET INTEGRITY statements.

The WebSphere FS 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 WebSphere FS 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 WebSphere FS system catalogs for later use. Dynamic SQL statements are unknown to WebSphere FS 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 WebSphere FS 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 ID (a specific user identifier), or by any relevant secondary authorization IDs (the identifiers for any relevant groups to which the primary authorization ID belongs), associated with the request from the client. If the privileges are held, then the section is passed to the SQL Runtime component for execution.

2.2.2.2 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 WebSphere FS 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 WebSphere FS 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 WebSphere FS 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 programming Application Program Interfaces (APIs) and utilities provided by WebSphere FS that do not use SQL statements to perform their specified actions. There exist a number of these APIs and utilities at both the WebSphere FS Instance level as well as at the individual database level within a WebSphere FS instance. Each API and utility provided by WebSphere FS has an assigned privilege or authority requirement as defined by WebSphere FS. The WebSphere FS 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 ID, or secondary authorization IDs where applicable, of the requestor.

2.2.4 Optional Features

DB2 includes some features that are enabled only when an applicable feature license is purchased and configured. Of particular note are the Database Partitioning Feature (DPF) feature. The TOE can be configured without this optional feature license. There are no specific claims, though the TOE continues to enforce its security policies whether it uses this partitioned feature or not.

2.2.4.1 DPF

As indicated above, DB2 includes a DPF feature. 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.

2.3 Product Features

The WebSphere FS TOE offers the following features:

- Ability to configure and allow uniform access to a wide range of data sources;
- 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;
- WebSphere FS provides support for both local and remote WebSphere FS clients.

2.4 Security Environment TOE Boundary

The TOE includes both physical and logical boundaries.

2.4.1 Physical Boundaries

As an application, the interfaces of WebSphere FS 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 IT environment.

WebSphere FS 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 WebSphere FS server. With the exception of those tools and utilities identified in the TOE's guidance documentation, these libraries and programs are outside the scope of the TOE.

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

WebSphere FS adds a set of interfaces beyond those of the core DB2 via a series of wrappers that are used to interface with other distributed servers. WebSphere FS supports the ability to proxy and coordinates requests between itself and other appropriately configured distributed data servers.

2.4.1.1 Wrappers

There are two interfaces of interest in relation to the wrappers used to access other data servers: the server to wrapper interface and there is the wrapper to client (i.e., of some other server) interface. Each wrapper that operates in a 'trusted' mode (meaning it operates in the same process context as the rest of the server) must be trustworthy and therefore would necessarily be part of the TSF – see Figure 3 WebSphere FS Trusted Mode. Wrappers that are 'fenced' – see Figure 4 WebSphere FS Fenced Mode – meaning that they operate in separate processes (Fenced Mode Process, FMP) are not part of the TSF. Note that the wrappers previously identified are part of the TOE since they are available with the product. However, the product is being evaluated with all wrappers (provided as part of the TOE/product or otherwise provided by the user) configured in fenced mode and hence they are outside the security boundary of the TSF.

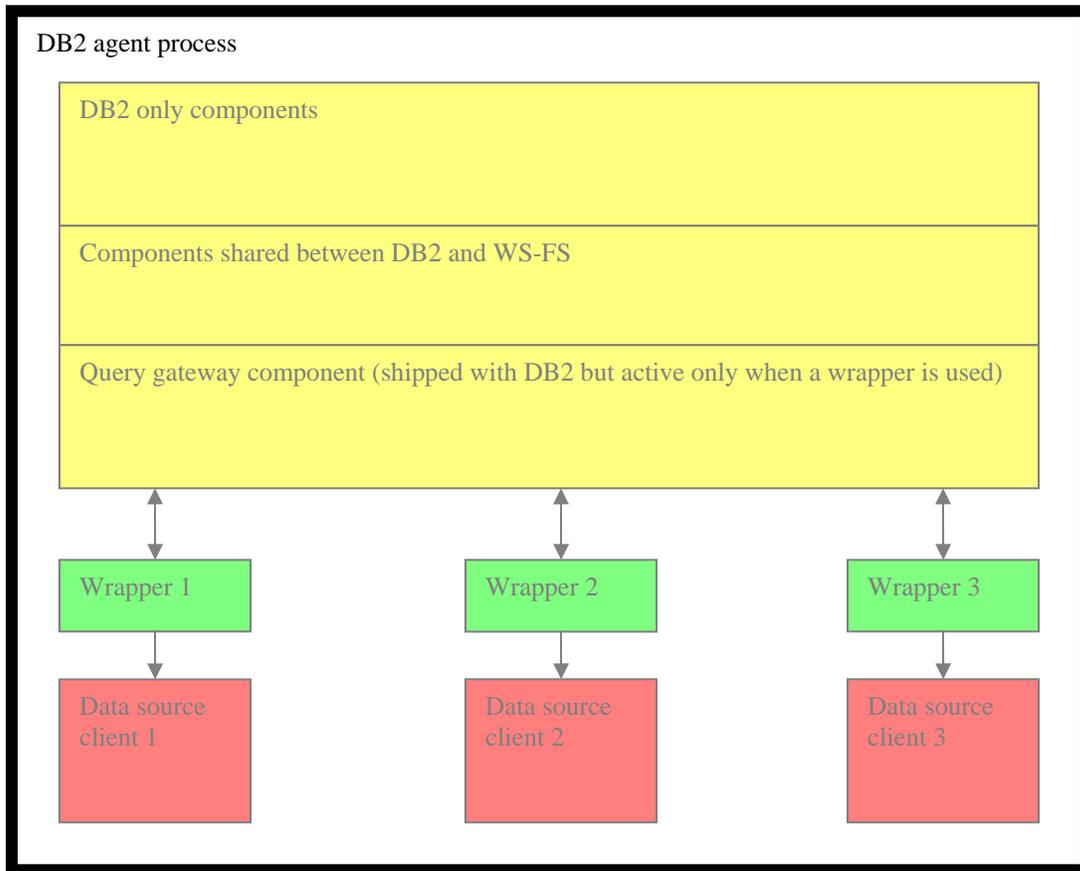


Figure 3 WebSphere FS Trusted Mode

In the Trusted-Mode case, the TSF boundary is the wrapper to client interface and in the Fenced-Mode case the TSF boundary is the server to wrapper interface. Since the evaluated configuration does not include Trusted-Mode wrappers, the specific TSF interface consists of the server to wrapper interface, while the wrapper to client interfaces of the wrappers specifically included in the TOE are non-TSF (i.e., not security relevant) TOE interfaces.

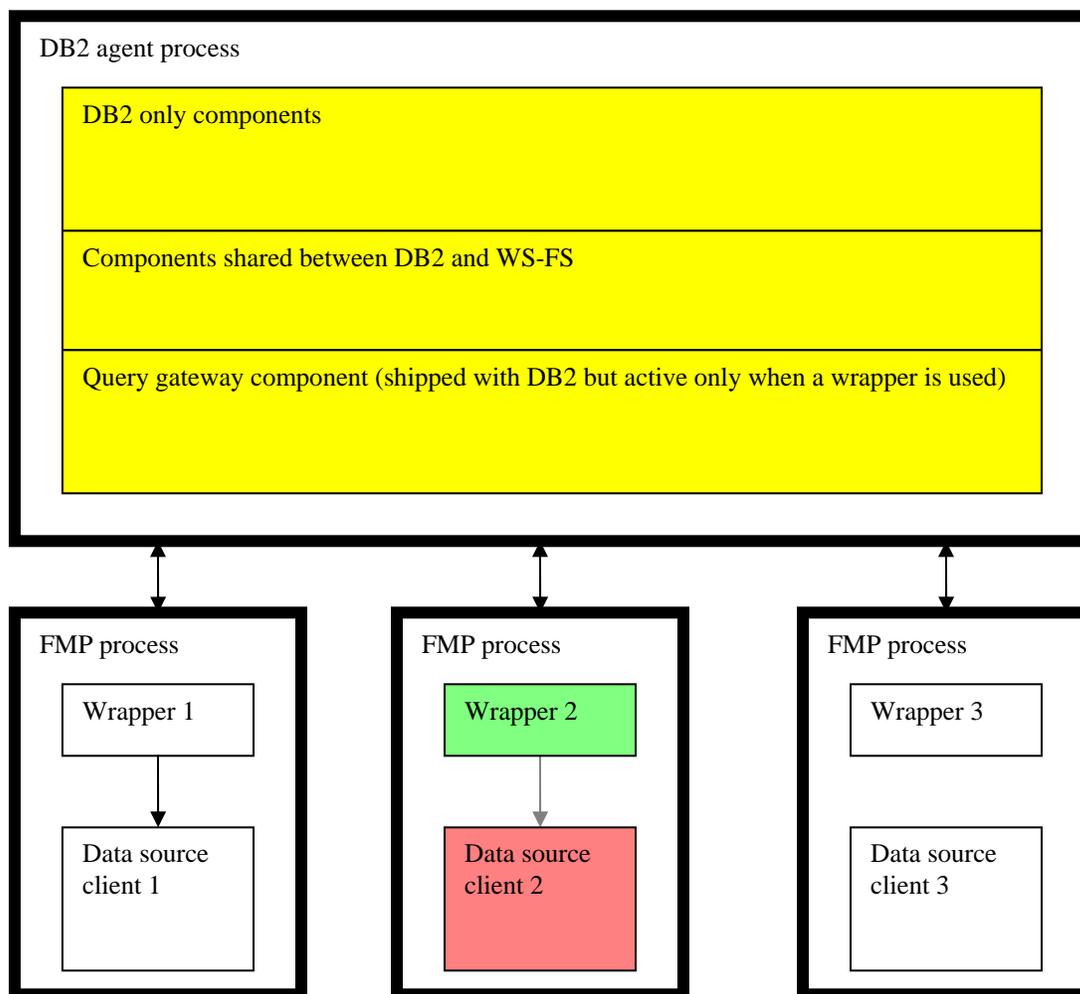


Figure 4 WebSphere FS Fenced Mode

2.4.2 Logical Boundaries

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

2.4.2.1 Security Audit

WebSphere FS 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.

2.4.2.2 Access Control

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

Note that while the term “roles” is used in this ST to distinguish authorized administrators from non-administrator users, WebSphere FS implements this concept using a variety of authorities and privileges. For this ST, references

to the “authorized administrator” role are implemented in WebSphere FS as any combination of the SYSADM, SYSCTRL, SYSMON, SYSMAINT, DBADM, or SECADM. References to the “user” role are implemented in WebSphere FS as any combination of lesser privileges (such as having the UPDATE or INSERT privilege on a specific database table).

In addition to using privileges and authorities to control access, WebSphere FS implements a Label-Based Access Control (LBAC) mechanism. The WebSphere FS SECADM 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 policies 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.

WebSphere FS also includes the ability to control who can pass requests to other data sources (via the wrappers) and who can access credentials that might be associated with other data sources.

2.4.2.3 Identification & Authentication

WebSphere FS requires all users to be identified and authenticated before allowing them access to WebSphere FS resources. The IT Environment performs the actual authentication and association of users with groups and passes the result to WebSphere FS. WebSphere FS subsequently enforces the result returned by the IT environment and uses the user identity and group memberships returned by the IT environment to associate privileges and authorities with the authenticated user.

2.4.2.4 Security Management

WebSphere FS includes the roles of authorized administrator and user, implemented using WebSphere FS authorities and privileges, and allows individual users to be assigned to those roles by virtue of group assignments in the IT environment. Management of the WebSphere FS TOE, including the ability to select and review audit records, is restricted to authorized administrators based on authorities. Management of WebSphere FS objects, including management of security labels, is restricted to those users that are assigned the appropriate privileges to do so.

2.4.2.5 TOE Protection

WebSphere FS executes within processes provided by the hosting IT environment. However, it is designed to not share its process space with non-TSF entities in order to ensure that TSF resources are protected. WebSphere FS has been designed so that each of its interfaces performs the necessary access checks before allowing access to WebSphere FS resources. Note that in the evaluated configuration there are no wrappers running in ‘trusted’ mode, so all wrappers are effectively outside of the TSF operating in their own processes.

3. Security Environment

Since DB2 embedded in WebSphere FS 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 WebSphere FS is a database system and not an operating system, some additional assumptions and security objectives have been assigned to the IT 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 system 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.

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 to access points such as terminals are assumed to be adequately protected.

A.PLATFORM

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

3.2 Threats

All security objectives, except for the non-IT security objectives for the environment, have been derived from the statement of Organizational Security Policy found in the following section. Non-IT security objectives for the environment have been drawn from the Secure Usage Assumptions detailed in Section 3.1. 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 the organizational security policies described below are drawn from DoD Manual 5200.28-M (Techniques and procedures for Implementing, Deactivating and Evaluating Resource Sharing ADP Systems) it applies 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 shall be held 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 and prevent a subject from causing information from being downgraded to a lower sensitivity.

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 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 record the security relevant actions of users of the TOE. The TSF must present this information 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 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 Non-IT security objectives for the 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 IT 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 TOE critical to security policy are protected from physical 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.

O.PLATFORM

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

4.2.2 IT security objectives for the environment

OE.AUTHORIZATION

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

OE.AUDITING

The IT Environment must record the security relevant actions of users of the IT Environment.

OE.RESIDUAL_INFORMATION

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

OE.MANAGE

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

OE.ENFORCEMENT

The IT Environment must be designed and implemented in a manner that ensures that it can protect the operational IT Environment of the TOE. The IT Environment must provide a reliable time source for the use of both the TOE and the IT Environment.

5. IT Security Requirements

The following sections define the security functional and assurance requirements for the TOE and its IT 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.

Note that this ST includes the security assurance requirement AVA_SOF.1, requiring strength of function analysis to demonstrate that each probabilistic or permutational security mechanism meets the stated strength of function (SOF) claim. However, the TOE does not include any probabilistic or permutational security mechanisms and, as a result, this ST makes no SOF claim.

5.1 TOE Security Functional Requirements

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

Table 1 TOE Functional Security Requirements

Security Functional Class	Security Functional Components
Security Audit (FAU)	FAU_GEN.1a 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
User Data Protection (FDP)	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.2a Full residual information protection
	FDP_ROL.1 Basic Rollback
Identification and authentication (FIA)	FIA_ATD.1a User attribute definition
	FIA_UAU.2a User authentication before any action (explicitly stated)
	FIA_UID.2a User identification before any action
	FIA_USB.1 User-subject binding
Security management (FMT)	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_REV.1a Revocation
	FMT_SMF.1a Specification of Management Functions
	FMT_SMR.1a Security roles
Protection of the TSF (FPT)	FPT_RVM.1a Non-bypassability of the TSP
	FPT_STM.1a Reliable Time Stamps (explicitly stated)

5.1.1 Security Audit (FAU)

FAU_GEN.1a Audit data generation

FAU_GEN.1a.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.1a.2 The TSF shall record within each audit record at least the following information:

- a) Date and time of the event, type of event, subject identity, and the outcome (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, [**no additional details**].

Table 2 Auditable Events for the TOE

Component	Event
FAU_GEN.1a	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.
FIA_UAU.2a	All use of the authentication mechanism
FIA_UID.2a	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
FMT_MSA.1a	All modifications of the values of security attributes
FMT_MSA.1b	All modifications of the values of security attributes
FMT_MTD.1a	All modifications to the values of TSF data
FMT_MTD.1b	All modifications to the values of TSF data
FMT_REV.1a	All modifications to the values of TSF data
FMT_SMF.1a	Use of the management functions

FAU_GEN.2 User identity association

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

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide [**authorised administrators**] with the capability to read [**all audit information**] from the audit records.

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

FAU_SAR.2 Restricted audit review

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.

FAU_SAR.3 Selectable audit review

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

FAU_SEL.1 Selective audit

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:
 a) [*event type*]
 b) [**success and/or failure**].

FAU_STG.3 Action in case of possible audit data loss

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

FAU_STG.4 Prevention of audit data loss

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

5.1.2 User Data Protection (FDP)

FDP_ACC.1 Subset access control

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, methods, wrappers, server definitions, user mappings, function templates, function mappings, index specifications, nicknames, and type mappings**].

FDP_ACF.1 Security attribute based access control

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
<i>Subjects</i>	
User	Authorization names and authorities
<i>Objects</i>	
Database Table space Wrapper Server definition User Mapping	Access control list ¹

¹ Access control lists assign privileges to users via authorization names.

Schema Table View Package Procedure Function Method Function template Function mapping Index specification Nickname Type mapping	Access control list Definer
---	--------------------------------

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 an authority defined by an authorized administrator can access objects 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 definer 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]**.

FDP_IFC. 1 Subset information flow control

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

FDP_IFF.2 Hierarchical security attributes

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]**².

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**
 - 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**

² 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.

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 **[no additional rules]**.
- FDP_IFF.2.4** The TSF shall provide the following **[only an appropriately privileged user can change security labels on users and columns or rows of LBAC protected tables]**.
- FDP_IFF.2.5** 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.2.6** The TSF shall explicitly deny an information flow based on the following rules: **[none]**.
- FDP_IFF.2.7** 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.

FDP_RIP.2a Full residual information protection

- FDP_RIP.2a.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.

FDP_ROL.1 Basic Rollback

- FDP_ROL.1.1** The TSF shall enforce **the [Discretionary Access Control Policy]** to permit the rollback of the **[operations that can be expressed as SQL]** on the **[embedded DB2 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]**.

5.1.3 Identification and authentication (FIA)

FIA_ATD.1a User attribute definition

- FIA_ATD.1a.1** The TSF shall maintain the following list of security attributes belonging to individual users: **[authorization names, authorities, security labels, and exemptions]**.

FIA_UAU.2a User authentication before any action (explicitly stated requirement)

- FIA_UAU.2a.1** The TSF shall require each user to be successfully authenticated using support from the IT environment before allowing any other TSF-mediated actions on behalf of that user.

FIA_UID.2a User identification before any action

- FIA_UID.2a.1** The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1 User-subject binding

- 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, security labels, and exemptions]**.
- 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 and the user's groups are assigned to the session as are authorities, security labels, and exemptions associated with those authorization names]**.
- 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]**.

5.1.4 Security management (FMT)

FMT_MOF.1 Management of security functions behaviour

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

FMT_MSA.1a Management of Security Attributes

- 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]** to **[users authorized by the Discretionary Access Control Rules]**.

FMT_MSA.1b Management of security attributes

- 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]**.

FMT_MSA.3a Static Attribute Initialization

- 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.

FMT_MSA.3b Static attribute initialization

- 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 **[no role]** to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1a Management of TSF data

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

FMT_MTD.1b Management of TSF data

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

FMT_REV.1a Revocation

FMT_REV.1a.1 The TSF shall restrict the ability to revoke security attributes associated with the [*objects*] within the TSC to **[users authorised to modify the Discretionary Access Control security attributes by the Discretionary Access Control policy and authorized administrators in the case of LBAC security attributes]**.

FMT_REV.1a.2 The TSF shall enforce the rules **[the access rights associated with an object shall be enforced when an access check is made]**.

FMT_SMF.1a Specification of Management Functions

FMT_SMF.1a.1 The TSF shall be capable of performing the following security management functions: **[start and stop auditing, select audited events, review the audit trail, creation of LBAC policies and granting of security labels]**.

FMT_SMR.1a Security Roles

FMT_SMR.1a.1 The TSF shall maintain the roles **[authorised administrator and user]**.

FMT_SMR.1a.2 The TSF shall be able to associate users with roles.

5.1.5 Protection of the TSF (FPT)

FPT_RVM.1a Non-bypassability of the TSP

FPT_RVM.1a.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_STM.1a Reliable Time Stamps (explicitly stated requirement)

FPT_STM.1a.1 The TSF shall be able to provide reliable time stamps based on information provided by the IT environment for its own use.

5.2 Security Requirements for the IT Environment

This section specifies the security requirements that are applicable to IT environment of the TOE.

Table 3 IT Environment Functional Security Requirements

Security Functional Class	Security Functional Components
Security Audit (FAU)	FAU_GEN.1b Audit data generation
	FAU_STG.1: Guarantees of Audit Data Availability
User Data Protection (FDP)	FDP_RIP.2b Full residual information protection
Identification and authentication (FIA)	FIA_ATD.1b User attribute definition
	FIA_SOS.1 Verification of secrets
	FIA_UAU.2b User authentication before any action
	FIA_UAU.7 Protected authentication feedback
Security management (FMT)	FIA_UID.2b User identification before any action
	FMT_MTD.1c Management of TSF data
	FMT_MTD.1d Management of TSF data
	FMT_MTD.1e Management of TSF data

Security Functional Class	Security Functional Components
Protection of the TSF (FPT)	FMT_REV.1b Revocation
	FMT_SMF.1b Specification of Management Functions
	FMT_SMR.1b Security Management Roles
	FPT_AMT.1 Abstract Machine Testing
	FPT_RVM.1b Reference Mediation
	FPT_SEP.1 Domain Separation
	FPT_STM.1b Reliable Time Stamps

5.2.1 Security Audit (FAU)

FAU_GEN.1b Audit data generation

FAU_GEN.1b.1 The ~~TSF~~**IT Environment** 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 4 Auditable Events**].

FAU_GEN.1b.2 The ~~TSF~~**IT Environment** shall record within each audit record at least the following information:

- a) Date and time of the event, type of event, subject identity, and the outcome (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, [**no additional details**].

Table 4 Auditable Events for the IT Environment

Component	Event
FAU_GEN.1b	Start-up and shutdown of the audit functions
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret
FIA_UAU.2b	All use of the authentication mechanism
FIA_UID.2b	All use of the user identification mechanism, including the identity provided during successful attempts
FMT_MTD.1c	All modifications to the values of TSF data
FMT_MTD.1d	All modifications to the values of TSF data
FMT_MTD.1e	All modifications to the values of TSF data
FMT_REV.1b	All attempts to revoke security attributes
FMT_SMR.1b	Modifications to the group of users that are part of a role
FPT_AMT.1	Execution of the tests of the underlying machine and the results of the test
FMT_REV.1b	All modifications to the values of TSF data
FMT_SMF.1b	Use of the management functions
FMT_SMR.1b	Modifications to the group of users that are part of a role
FPT_STM.1b	Changes to the time

FAU_STG.1Guarantees of Audit Data Availability

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

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

5.2.2 User Data Protection (FDP)

FDP_RIP.2b Full residual information protection

FDP_RIP.2b.1 The ~~TSP~~-IT Environment shall ensure that any previous information content of a resource is made unavailable upon the [*allocation of the resource to*] all objects.

5.2.3 Identification and authentication (FIA)

FIA_ATD.1b User attribute definition

FIA_ATD.1b.1 The ~~TSP~~-IT Environment shall maintain the following list of security attributes belonging to individual users: [**user identifier, group memberships, roles, and authentication data**].

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The ~~TSP~~-IT Environment shall provide a mechanism to verify that secrets meet [**the following**]

- a) for each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000;**
- b) for multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000; and**
- c) any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.**

FIA_UAU.2b User authentication before any action

FIA_UAU.2b.1 The ~~TSP~~-IT Environment shall require each user to be successfully authenticated before allowing any other TSP-mediated actions on behalf of that user **and when requested by the TOE**.

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The ~~TSP~~-IT Environment shall provide only [**obscured feedback**] to the user while the authentication is in progress.

FIA_UID.2b User identification before any action

FIA_UID.2b.1 The ~~TSP~~-IT Environment shall require each user to identify itself before allowing any other TSP-mediated actions on behalf of that user.

5.2.4 Security management (FMT)

FMT_MTD.1c Management of TSP data

FMT_MTD.1c.1 The ~~TSP~~-IT Environment shall restrict the ability to [*modify and initialize*] the [**user security attributes other than authentication data**] to [**authorised administrators**].

FMT_MTD.1d Management of TSP data

FMT_MTD.1d.1 The ~~TSP~~-IT Environment shall restrict the ability to [*initialize*] the [**authentication data**] to [**authorised administrators**].

FMT_MTD.1e Management of TSP data

FMT_MTD.1e.1 The ~~TSP~~-IT Environment shall restrict the ability to [*modify*] the [**authentication data**] to [**the following: authorised administrators and users authorised to modify their own authentication data**].

FMT_REV.1b Revocation

FMT_REV.1b.1 The ~~TSP~~-IT Environment shall restrict the ability to revoke security attributes associated with the [users] within the TSC to [authorised administrators].

FMT_REV.1b.2 The ~~TSP~~-IT Environment shall enforce the rules [the attributes associated with users shall be applied when a user is identified and authenticated].

FMT_SMF.1b Specification of Management Functions

FMT_SMF.1b.1 The ~~TSP~~-IT Environment shall be capable of performing the following security management functions: [create, modify, and delete user accounts].

FMT_SMR.1b Security Management Roles

FMT_SMR.1b.1 The ~~TSP~~-IT Environment shall maintain the roles [authorised administrator].

FMT_SMR.1b.2 The ~~TSP~~-IT Environment shall be able to associate users with roles.

5.2.5 Protection of the TSF (FPT)

FPT_AMT.1 Abstract Machine Testing

FPT_AMT.1.1 The ~~TSP~~-IT Environment shall run a suite of tests [*at the request of an authorised user*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlie the TSF.

FPT_RVM.1b Reference Mediation

FPT_RVM.1b.1 The ~~TSP~~-IT Environment shall ensure the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1 Domain Separation

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

FPT_SEP.1.2 The ~~TSP~~-IT Environment shall enforce separation between the security domains of subjects in the TSC.

FPT_STM.1b Reliable Time Stamps

FPT_STM.1b.1 The ~~TSP~~-IT Environment shall be able to provide reliable time stamps for its own use **and for use by its subjects**.

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 5 Assurance Requirements (EAL 4 augmented)

Assurance Class	Assurance Components
Configuration Management (ACM)	ACM_AUT.1 Partial CM automation
	ACM_CAP.4 Generation support and acceptance procedures
	ACM_SCP.2 Problem tracking CM coverage
Delivery and Operation (ADO)	ADO_DEL.2 Detection of modification
	ADO_IGS.1 Installation, generation, and start-up procedures
Development (ADV)	ADV_FSP.2 Fully defined external interfaces
	ADV_HLD.2 Security enforcing high-level design
	ADV_IMP.1 Subset of the implementation of the TSF
	ADV_LLD.1 Descriptive low-level design
	ADV_RCR.1 Informal correspondence demonstration
	ADV_SPM.1 Informal TOE security policy model
Guidance Documents (AGD)	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Life cycle support (ALC)	ALC_DVS.1 Identification of security measures
	ALC_FLR.1 Basic flaw remediation
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
Tests (ATE)	ATE_COV.2 Analysis of Coverage
	ATE_DPT.1 Testing: high-level design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
Vulnerability assessment (AVA)	AVA_MSU.2 Validation of analysis
	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.2 Independent vulnerability analysis

5.3.1 Configuration Management (ACM)

ACM_AUT.1 Partial CM automation

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

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

ACM_CAP.4 Generation support and acceptance procedures

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labelled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list and a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items.

ACM_CAP.4.8C The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

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

ACM_SCP.2 Problem tracking CM coverage

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

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

5.3.2 Delivery and Operation (ADO)

ADO_DEL.2 Detection of modification

- ADO_DEL.2.1D** The developer shall document procedures for delivery of the TOE or parts of it to the user.
- ADO_DEL.2.2D** The developer shall use the delivery procedures.
- ADO_DEL.2.1C** The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
- ADO_DEL.2.2C** The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
- ADO_DEL.2.3C** The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.
- ADO_DEL.2.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1 Installation, generation, and start-up procedures

- ADO_IGS.1.1D** The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
- ADO_IGS.1.1C** The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.
- ADO_IGS.1.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADO_IGS.1.2E** The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

ADV_FSP.2 Fully defined external interfaces

- ADV_FSP.2.1D** The developer shall provide a functional specification.
- ADV_FSP.2.1C** The functional specification shall describe the TSF and its external interfaces using an informal style.
- ADV_FSP.2.2C** The functional specification shall be internally consistent.
- ADV_FSP.2.3C** The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
- ADV_FSP.2.4C** The functional specification shall completely represent the TSF.
- ADV_FSP.2.5C** The functional specification shall include rationale that the TSF is completely represented.
- ADV_FSP.2.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.2 Security enforcing high-level design

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

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

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_IMP.1 Subset of the implementation of the TSF

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

ADV_IMP.1.1C The implementation representation shall unambiguously 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 internally consistent.

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

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

ADV_LLD.1 Descriptive low-level design

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

- ADV_LLD.1.2C** The low-level design shall be internally consistent.
- ADV_LLD.1.3C** The low-level design shall describe the TSF in terms of modules.
- ADV_LLD.1.4C** The low-level design shall describe the purpose of each module.
- ADV_LLD.1.5C** The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.
- ADV_LLD.1.6C** The low-level design shall describe how each TSP-enforcing function is provided.
- ADV_LLD.1.7C** The low-level design shall identify all interfaces to the modules of the TSF.
- ADV_LLD.1.8C** The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
- ADV_LLD.1.9C** The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.
- ADV_LLD.1.10C** The low-level design shall describe the separation of the TOE into TSP enforcing and other modules.
- ADV_LLD.1.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADV_LLD.1.2E** The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

- ADV_RCR.1.1D** The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
- ADV_RCR.1.1C** For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
- ADV_RCR.1.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_SPM.1 Informal TOE security policy model

- ADV_SPM.1.1D** The developer shall provide a TSP model.
- ADV_SPM.1.2D** The developer shall demonstrate correspondence between the functional specification and the TSP model.
- ADV_SPM.1.1C** The TSP model shall be informal.
- ADV_SPM.1.2C** The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
- ADV_SPM.1.3C** The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

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

5.3.4 Guidance Documents (AGD)

AGD_ADM.1 Administrator guidance

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

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

AGD_USR.1 User guidance

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

- AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.
- AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
- AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life Cycle Support (ALC)

ALC_DVS.1 Identification of security measures

- 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.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.
- 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.

ALC_FLR.1 Basic flaw remediation

- ALC_FLR.1.1D The developer shall provide 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.

ALC_LCD.1 Developer defined life-cycle model

- 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.

ALC_TAT.1 Well-defined development tools

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools 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.

5.3.6 Security Testing (ATE)

ATE_COV.2 Analysis of Coverage

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 identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

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

ATE_DPT.1 Testing: high-level design

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

ATE_DPT.1.2E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

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, test procedure descriptions, expected test results and actual test results.

- ATE_FUN.1.2C** The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
- ATE_FUN.1.3C** The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
- ATE_FUN.1.4C** The expected test results shall show the anticipated outputs from a successful execution of the tests.
- ATE_FUN.1.5C** The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
- ATE_FUN.1.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing – sample

- 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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
- ATE_IND.2.3E** The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability Assessment (AVA)

AVA_MSU.2 Validation of analysis

- AVA_MSU.2.1D** The developer shall provide guidance documentation.
- AVA_MSU.2.2D** The developer shall document an analysis of the guidance documentation.
- AVA_MSU.2.1C** The guidance documentation 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.
- AVA_MSU.2.2C** The guidance documentation shall be complete, clear, consistent and reasonable.
- AVA_MSU.2.3C** The guidance documentation shall list all assumptions about the intended environment.
- AVA_MSU.2.4C** The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
- AVA_MSU.2.5C** The analysis documentation shall demonstrate that the guidance documentation is complete.

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

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_SOF.1 Strength of TOE security function evaluation

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

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

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

AVA_VLA.2 Independent vulnerability analysis

AVA_VLA.2.1D The developer shall perform a vulnerability analysis.

AVA_VLA.2.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.2.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

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

AVA_VLA.2.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.2.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.2.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low 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 WebSphere FS audit facility acts at an instance level, recording all instance level activities and database level activities. The audit log (db2audit.log) and the audit configuration file (db2audit.cfg) are located in the instance's security subdirectory.

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

Authorized users of the audit facility tool can control the following actions within the audit facility:

- Start recording auditable events within the WebSphere FS instance.
- Stop recording auditable events within the WebSphere FS 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 for one of two reasons: in preparation for analysis of log records or in preparation for pruning of log records.
- Prune audit records from the current audit log.

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 WebSphere FS 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 associated with the audit record

WebSphere FS provides only the authorized administrator (based on their role) 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 an authorized administrator, in ASCII format and loaded into a DB2 relational table for a rich set of query capabilities.

The authorized administrator can configure WebSphere FS to audit any or all of the available audit categories:

- Audit (AUDIT) – Generates records when audit settings are changed or when the audit log is accessed.
- Authorization Checking (CHECKING) – Generates records during authorization checking of attempts to access, transfer, or manipulate WebSphere FS objects or functions.
- Object Maintenance (OBJMAINT) – Generates records when creating or dropping data objects and LBAC policies.
- Security Maintenance (SECMAINT) – Generates records when granting or revoking: object or database privileges or authorities, 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 authorized administrator can configure WebSphere FS to audit only successful, only failed, or successful and failed audit events, configured per audit event type.

WebSphere FS does not provide any ability within the TOE to modify the audit records. Due to role restrictions, WebSphere FS offers functions allowing only an authorized administrator 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, an authorized administrator can configure WebSphere FS 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.1a Audit data generation – WebSphere FS 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.

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

FAU_SAR.1 Audit review – WebSphere FS fulfills this requirement by providing an interface for the review of audit records.

FAU_SAR.2 Restricted audit review – WebSphere FS fulfills this requirement by ensuring that the user is an authorized administrator (per their role) before allowing access to the audit records.

FAU_SAR.3 Selectable audit review – WebSphere FS 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 – WebSphere FS fulfills this requirement by allowing an authorized administrator to configure WebSphere FS to audit any or all of the available audit categories as well as successful and/or failed events.

FAU_STG.3 Action in case of possible audit data loss – WebSphere FS 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 – WebSphere FS fulfills this requirement by allowing an authorized administrator to configure WebSphere FS 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 WebSphere FS obtains information about an authenticated WebSphere FS 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.

WebSphere FS logically associates access control lists and a ‘definer’³ 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 to which the user belongs, are compared with access control list entries to find matches. Based on this comparison, WebSphere FS identifies available permissions that indicate whether to allow the requested access.

There are two types of permissions managed by WebSphere FS: 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; system authorities are associated with group membership, and are stored in the database manager configuration file for a given instance.

Groups provide a 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 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 commands and utilities), but is not considered for static SQL. 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.

WebSphere FS 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.

³ Note that a definer is associated only with the following database objects: Schema, Table, View, Package, Procedure, Function, Method, Function template, Function mapping, Index specification, Nickname, and Type mapping.

- Procedure, function, and method privileges, which involve actions on routines such as functions, procedures and methods.
- Server definition privileges, which involve actions related to setting up a passthrough of SQL statements to another server.

With regard to the objects introduced by WebSphere FS that are not included in DB2:

- Function templates, index specifications, and nicknames are similar to DB2 functions, indexes, and tables, respectively. They require the same privileges to create, drop, or alter and they have the same access privileges. For example, in order to read from a WebSphere FS nicknames or a DB2 table, you need SELECT privilege.
- Server definitions are the only WebSphere FS-specific objects that have their own specific access privilege (PASSTHRU) as indicated above. This privilege gives the user the ability to issue the SQL statement 'SET PASSTHRU server_name'. Issuing that statement specifies that DB2 is to send subsequent SQL statements directly to the specified server without going through the DB2 compiler. To turn that off the user can issue SQL statement 'SET PASSTHRU RESET'. Note that server definitions serve to define remote data servers.
- The other objects (function mappings, type mappings, user mappings, and wrappers) require privileges to create, drop, or alter, but they have no specific access privileges that can be granted or revoked. Note that user mappings serve to define user credentials for remote data sources.

In order for a user to access (including create and destroy, except for remote server and credential accesses) any of the objects, including access to information on a remote data source via a protected object (e.g., nickname), 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 SYSADM authority, for example, implicitly has many privileges. Similarly, an object definer is not subject to the privilege restrictions on the objects they defined (i.e., created).

Authorized administrators have the ability to grant authorities and privileges to other users and groups. 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.

In addition to controlling access using permissions, when a LBAC license is properly configured, authorized 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, otherwise 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. There are two sets of three rules that determine the allowed access:

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.

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.

In addition to the rules cited above, WebSphere FS 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 – WebSphere FS fulfills this requirement by associating privileges with all operations applicable to each identified WebSphere FS object and requiring that a user have the privilege or authority when attempting to perform the corresponding operation.

FDP_ACF.1 Security attribute based access control – WebSphere FS fulfills this requirement by associating privileges with all operations applicable to each identified WebSphere FS 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 – WebSphere FS 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 – WebSphere FS fulfills this requirements by enforcing the LBAC information flows rules as summarized above.

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

FDP_ROL.1 Basic Rollback – WebSphere FS 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 WebSphere FS without a user ID and password while logged on to the WebSphere FS host operating system (i.e., IT Environment), WebSphere FS 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 WebSphere FS uses to identify users and groups.

When a user attempts to access WebSphere FS remotely (i.e., while not logged onto the WebSphere FS host operating system), they must provide a user identity and password. The user identity and password are provided to the WebSphere FS host operating system for authentication. If the WebSphere FS host operating system determines that the user identity exists and the password is valid⁴, it will respond to WebSphere FS with the authenticated user identity and any applicable group memberships. Otherwise, it will return a failed result that will cause WebSphere FS to reject the request.

Once authenticated:

- The user must be identified to WebSphere FS 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, WebSphere FS authids are usually derived by transforming the user ID to all uppercase letters. If the resulting authid fails to follow any WebSphere FS naming conventions (e.g., allowed characters or size), then WebSphere FS will reject the access request.
- A list of groups to which the user belongs is obtained. Groups are WebSphere FS host operating system constructs that must also map to WebSphere FS authorization names. This mapping is done in a method similar to that used for user IDs. However, any groups that fail to follow WebSphere FS naming conventions will be ignored, but the access attempt will not be rejected on this basis.

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 memberships are also used for access control. Once the identification and authentication process successfully yields one or more authids, WebSphere FS instantiates a session with those authids for WebSphere FS to allow mediated WebSphere FS operations. Furthermore, WebSphere FS associates specific authorities with the authids available to the user – this is also known as “Authorization.”

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

FIA_ATD.1a User attribute definition – WebSphere FS fulfills this requirement by maintaining a correspondence between authids, authorities, security labels, and exemptions associated with users.

FIA_UAU.2a User authentication before any action – WebSphere FS fulfills this requirement by rejecting access to WebSphere FS resources when the user cannot be authenticated using support from the IT environment. Note that requirements are also instantiated in the IT environment such that it similarly protects itself and ensures appropriate strength of this mechanism,

FIA_UID.2a User identification before any action – WebSphere FS fulfills this requirement by rejecting access to WebSphere FS resources when the user cannot be identified.

FIA_USB.1 User-subject binding – WebSphere FS fulfills this requirement by associating authids as well as authorities, security labels, and exemptions with user sessions.

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. WebSphere FS 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 WebSphere FS 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

⁴ Note that in addition to the password being correct, it must also not be expired. Additionally, some host operating systems have additional identification and authentication conditions that can cause additional authentication failures (account locked, time of day restrictions, workstation restrictions, etc.).

control access to the database manager and its database objects. Users can access (including attribute modification) 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 WebSphere FS 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 WebSphere FS and cannot be modified by any user. Note that of all of the protected WebSphere FS 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., an authorized administrator) and only when the TOE has been properly configured with a LBAC feature license. Note that 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). WebSphere FS provides the administrator functions 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.

For privileges associated with a user identity or with a group membership, the privilege may be revoked from the user. In this case, 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 an authorized administrator. Changes to access rights associated with an object are not effective until the next access check that would normally be required.

As described in Section 6.1.1, Security Audit, WebSphere FS provides authorized administrators with the functions necessary to start and stop the audit security function, as well as the tools necessary 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.

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 authorized administrator.

FMT_MSA.1a Management of Security Attributes – WebSphere FS fulfills this requirement by allowing only users with the appropriate privilege to modify the Discretionary Access Control security attributes of any WebSphere FS object.

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.

FMT_MSA.3a Static Attribute Initialization – WebSphere FS 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.

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

FMT_MTD.1b Management of TSF data – WebSphere FS fulfills this requirement by allowing only authorized administrators to access the audit trail for review. Note that WebSphere FS does not provide any capability to modify audit records.

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

FMT_SMF.1a Specification of Management Functions – WebSphere FS fulfills this requirement by providing functions that allow an authorized administrator to start, stop, configure the audit security functions as well as the ability to review audit records, create LBAC policies, and grant LBAC security labels.

FMT_SMR.1a Security Management Roles – WebSphere FS fulfills this requirement by defining authorized administrator and user roles.

6.1.5 TOE Protection

WebSphere FS is designed to operate within a set of processes provided by the hosting operating system. WebSphere FS does not support the ability to share its processes with non-TOE entities. Furthermore, WebSphere FS 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 WebSphere FS, its protection mechanisms could be bypassed through the underlying environment should the assumptions (e.g., A.Platform) and requirements (e.g., FPT_SEP.1) for the IT environment not be fulfilled. Note that determination of fulfillment of those assumptions and IT environment requirements is not within the scope of the TOE.

WebSphere FS has been designed to implement a number of WebSphere FS-specific objects and functions. Each WebSphere FS object and function is available via interfaces provided by WebSphere FS, 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.

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

The TOE Protection security function satisfies the following security requirements:

FPT_RVM.1a Reference Mediation – WebSphere FS fulfills this requirement by making sure that all applicable access checks are made by each of its interfaces before allowing access to WebSphere FS resources.

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

6.2 TOE Security Assurance Measures

The following assurance measures are applied to satisfy the Common Criteria EAL 4 (augmented with ALC_FLR.1) assurance requirements:

- Process Assurance,
- Delivery and Guidance,
- Design Documentation,
- Tests, and
- Vulnerability Assessment.

6.2.1 Process Assurance

6.2.1.1 Configuration Management

The configuration management measures applied by IBM ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. IBM ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated configuration item modifications are properly controlled. IBM performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, the CM documentation, and security flaws. These activities are documented in:

- IBM WebSphere FS Configuration Management Plan

6.2.1.2 Life cycle support

IBM ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. IBM includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure operation of the TOE. IBM achieves this through the use of a documented model of the TOE life-cycle and well-defined development tools that yield consistent and predictable results. IBM has procedures for accepting and addressing identified flaws, including tracking, describing, correcting, and taking other remedial actions such as producing guidance related to such flaws. These procedures are documented in:

- IBM WebSphere FS Life-Cycle Plan

The Process Assurance measures satisfy the following assurance requirements:

- ACM_AUT.1
- ACM_CAP.4,
- ACM_SCP.2,
- ALC_DVS.1,
- ALC_FLR.1,
- ALC_LCD.1, and
- ALC_TAT.1.

6.2.2 Delivery and Guidance

IBM 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.

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 WebSphere FS in accordance with the evaluated configuration.

All of the delivery and installation procedures, as well as administrator and user guidance is documented in:

- IBM WebSphere FS Delivery Procedures document
- IBM WebSphere FS Common Criteria Certification: Administration and User Documentation
- IBM WebSphere FS Common Criteria Certification: Installing WebSphere FS

The Delivery and Guidance assurance measure satisfies the following assurance requirements:

- ADO_DEL.2,
- ADO_IGS.1,
- AGD_ADM.1, and
- AGD_USR.1.

6.2.3 Development

The Design Documentation security assurance measure satisfies the following security assurance requirement:

- ADV_FSP.2: The IBM WebSphere FS Functional Specification fully describes all interfaces to the TSF.
- ADV_HLD.2: The IBM WebSphere FS High-level Design satisfies the requirement for decomposing the TOE into subsystems and fully describes each subsystem, including inter-subsystem interfaces.
- ADV_LLD.1: The IBM WebSphere FS Low-level Design satisfies the requirement to decompose each subsystem into modules and fully describes each module.
- ADV_IMP.1: A subset of the source code used to generate the TOE satisfies this requirement.
- ADV_RCR.1: Most of the correspondence between the various design documentation is implicit to the way in which the documentation is structured. The way that this correspondence is evident within the design documentation is:
 - ST-TSS to FSP: The IBM WebSphere FS Functional Specification describes how the interfaces correspond with the security functions in the ST.
 - FSP to HLD: The IBM WebSphere FS High-level Design describes how the various security behavior in the IBM WebSphere FS Functional Specification are further refined.
 - HLD to LLD: The IBM WebSphere FS Low-level Design describes how the various security behavior in the IBM WebSphere FS High-level Design are further refined.
 - LLD to IMP: The IBM WebSphere FS Low-level Design also serves to correspond modules with their specific implementations.
- ADV_SPM.1: The IBM WebSphere FS Security Model models the entities and rules related to the policies for identification and authentication, audit, and all of the information flow policies. Additionally, correspondence with the IBM WebSphere FS Functional Specification is described.

6.2.4 Tests

The Tests assurance measure satisfies the following assurance requirements:

- ATE_COV.2: The test case descriptions (in the IBM WebSphere FS Functional Specification) describe the test cases for each of the security-relevant interfaces of the TOE. The descriptions indicate which tests are used to satisfy the test cases identified for each interface.
- ATE_DPT.1: The test case descriptions (in the IBM WebSphere FS High-level Design) include more detailed test case descriptions that demonstrate that all of the corresponding interfaces are appropriately exercised.
- ATE_FUN.1: The IBM WebSphere FS Test Plan, describes the security functions to be tested, how to successfully test all of them, the expected results, and the actual test results after exercising all of the tests.
- ATE_IND.2: The TOE and test documentation will be available for independent testing.

6.2.5 Vulnerability Assessment

6.2.5.1 Evaluation of Misuse

The IBM WebSphere FS Common Criteria Certification Guide describes the operation of WebSphere FS and how to maintain a secure state. This guide also describes all operating assumptions and security requirements outside the scope of control of the TOE. It has been developed to serve as complete, clear, consistent, and reasonable administrator and user references. This guide is documented in:

- IBM WebSphere FS Common Criteria Certification: Administration and User Documentation

The misuse analysis shows that the administrative and user guidance completely addresses managing the TOE in a secure configuration.

- IBM WebSphere FS Misuse Analysis

6.2.5.2 Strength of TOE Security Functions

There are no strength of function claims associated with the WebSphere FS TOE. Therefore, there is no strength of function analysis document.

6.2.5.3 Vulnerability Analysis

IBM performs vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE. The vulnerability analysis is documented in:

- IBM WebSphere FS Vulnerability Analysis

The Vulnerability Assessment assurance measure satisfies the following assurance requirements:

- AVA_MSU.2,
- AVA_SOF.1, and
- AVA_VLA.2.

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 6 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
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS
	O.RESIDUAL_INFORMATION
	OE.RESIDUAL_INFORMATION
	O.ROLLBACK
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
P.ACCOUNTABILITY	O.AUDITING
	OE.AUDITING
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
P.CLASSIFICATION	O.MANDATORY_ACCESS
	O.RESIDUAL_INFORMATION
	O.MANAGE
	O.ENFORCEMENT

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 IT 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.

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.

P.ACCOUNTABILITY

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

This policy is primarily realized by the O.AUDITING and OE.AUDITING objectives by requiring that actions are 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.

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 and prevent a subject from causing information from being downgraded to a lower sensitivity.

This policy is implemented by the O.MANDATORY_ACCESS objective. The O.RESIDUAL_INFORMATION objective ensures that information will not be 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.

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 7 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 system 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 IT Environment 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 to access points such as terminals are assumed to be adequately protected.

This is addressed by O.PHYSICAL which ensures that those parts of the TOE critical to security policy are protected from physical 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 IT Environment underlying the TOE is assumed to fulfill the requirements for the IT Environment described in this Security Target. It is also assumed that the IT Environment will provide a suitable operational environment for the TOE where the TOE will be able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled.

This is addressed by O.PLATFORM that basically reiterates the assumption to expect the IT 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.

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 functional requirements that support other requirements, as detailed in the following table.

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	

FPT_RVM.1	Prevent other security functional requirements from being bypassed
FPT_SEP.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 8 Mapping of Security Objectives to Functional Components

Security Objective	Functional Component
O.AUTHORIZATION	FIA_ATD.1a
	FIA_UAU.2a
	FIA_UID.2a
OE.AUTHORIZATION	FIA_ATD.1b
	FIA_SOS.1
	FIA_UAU.2b
	FIA_UAU.7
	FIA_UID.2b
	FMT_MTD.1d
O.DISCRETIONARY_ACCESS	FMT_MTD.1e
	FDP_ACC.1
	FDP_ACF.1
	FIA_ATD.1a
	FIA_USB.1
	FMT_MSA.1a
O.MANDATORY_ACCESS	FMT_MSA.3a
	FMT_REV.1a
	FDP_IFC.1
	FDP_IFF.2
	FIA_ATD.1a
	FIA_USB.1
	FMT_MOF.1
O.AUDITING	FMT_MSA.1b
	FMT_MSA.3b
	FAU_GEN.1a
	FAU_GEN.2
	FAU_SAR.1
	FAU_SAR.2
	FAU_SAR.3
	FAU_SEL.1
	FAU_STG.1
	FAU_STG.3
	FAU_STG.4
	FIA_USB.1
	FMT_MTD.1a
FMT_MTD.1b	
FMT_SMF.1a	
FPT_STM.1a	

Security Objective	Functional Component
OE.AUDITING	FAU_GEN.1b
	FPT_STM.1b
O.RESIDUAL_INFORMATION	FDP_RIP.2a
OE.RESIDUAL_INFORMATION	FDP_RIP.2b
O.ROLLBACK	FDP_ROL.1
O.MANAGE	FAU_SAR.1
	FAU_SAR.3
	FAU_SEL.1
	FAU_STG.3
	FAU_STG.4
	FMT_MTD.1a
	FMT_MTD.1b
	FMT_SMF.1a
	FMT_SMR.1a
OE.MANAGE	FMT_MTD.1c
	FMT_MTD.1d
	FMT_MTD.1e
	FMT_REV.1b
	FMT_SMF.1b
	FMT_SMR.1b
O.ENFORCEMENT	FPT_RVM.1a
OE.ENFORCEMENT	FPT_AMT.1
	FPT_RVM.1b
	FPT_SEP.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.2a], authenticated [FIA_UAU.2a], and associated with available authorities and privileges [FIA_ATD.1a] before they can access the TOE and the resources it protects.

OE.AUTHORIZATION

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

Users must be identified [FIA_UID.2b], authenticated [FIA_UAU.2b], and associated with available roles and privileges [FIA_ATD.1b] before they can access the IT Environment and the resources it protects. Furthermore, the authentication data must be protected [FIA_UAU.7, FMT_MTD.1d, FMT_MTD.1e] and the authentication mechanism must have suitable strength [FIA_SOS.1].

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 be able to revoke that access [FMT_REV.1a]. 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.1a, FIA_USB.1].

O.AUDITING

The TSF must record the security relevant actions of users of the TOE. The TSF must present this information 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_STG.1, 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.1a]. An authorized administrator must be able to review [FAU_SAR.1] and manage [FAU_STG.3, FMT_MTD.1a, FMT_MTD.1b, FMT_SMF.1a] the audit trail.

OE.AUDITING

The IT Environment must record the security relevant actions of users of the IT Environment.

Security-relevant actions must be defined and auditable in the IT Environment [FAU_GEN.1b] and the audit records must have reliable time stamps [FPT_STM.1b].

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.2a].

OE.RESIDUAL_INFORMATION

The IT Environment 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 purged prior to the reuse of the object containing the residual information [FDP_RIP.2b].

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 an embedded DB2 protected resource 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.1a]. 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_MTD.1a, FMT_MTD.1b, FMT_SMF.1a].

OE.MANAGE

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

The IT Environment must provide for an authorized administrator to manage the IT Environment [FMT_SMR.1b]. The administrator must be able to administer user accounts [FMT_MTD.1c, FMT_MTD.1d, FMT_MTD.1e, FMT_REV.1b, FMT_SMF.1b].

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 [FPT_RVM.1a]. 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.

OE.ENFORCEMENT

The IT Environment must be designed and implemented in a manner that ensures that it can protect the operational IT Environment of the TOE. The IT Environment must provide a reliable time source for the use of both the TOE and the IT Environment.

The IT Environment must make and enforce the decisions of its security policies [FPT_RVM.1b]. It must be protected from interference that would prevent it from performing its security functions [FPT_SEP.1]. Additionally, the IT Environment must provide the capability to demonstrate correct operation of the underlying abstract machine [FPT_AMT.1]. The IT Environment must also supply reliable time stamps [FPT_STM.1b].

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.

Note that in the left column TOE security functional requirements are identified normally, IT environment security functional requirements are *italicized*, and assurance requirements are underlined.

Table 9 TOE Security Functional Requirement Dependencies

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1a	FPT_STM.1	FPT_STM.1a/ <i>FPT_STM.1b</i>
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1a and FIA_UID.2a
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1a
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.1a and FMT_MTD.1b
FAU_STG.3	FAU_STG.1	<i>FAU_STG.1</i>
FAU_STG.4	FAU_STG.1	<i>FAU_STG.1</i>
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.2a	none	none
FDP_ROL.1	none	none
FIA_ATD.1a	none	none
FIA_UAU.2a	FIA_UID.1	FIA_UID.2a/ <i>FIA_UID.2b</i>
FIA_UID.2a	none	none
FIA_USB.1	FIA_ATD.1	FIA_ATD.1a
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1a and FMT_SMF.1a
FMT_MSA.1a	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1a and FMT_SMF.1a and FDP_ACC.1
FMT_MSA.1b	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1a and FMT_SMF.1a and FDP_IFC.1
FMT_MSA.3a	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1a and FMT_SMR.1
FMT_MSA.3b	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1b and FMT_SMR.1
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1a and FMT_SMF.1a
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1a and FMT_SMF.1a
FMT_REV.1a	FMT_SMR.1	FMT_SMR.1a
FMT_SMF.1a	none	none
FMT_SMR.1a	FIA_UID.1	FIA_UID.2a
FPT_RVM.1a	none	none
FPT_STM.1a	none	none
FAU_GEN.1b	FPT_STM.1	<i>FPT_STM.1b</i>
FAU_STG.1	FAU_GEN.1	FAU_GEN.1a/ <i>FAU_GEN.1b</i>
FDP_RIP.2b	none	none
FIA_ATD.1b	none	none
FIA_SOS.1	none	none
FIA_UAU.2b	FIA_UID.1	<i>FIA_UID.2b</i>
FIA_UAU.7	FIA_UAU.1	<i>FIA_UAU.2b</i>
FIA_UID.2b	none	none
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	<i>FMT_SMR.1b</i> and <i>FMT_SMF.1b</i>
FMT_MTD.1d	FMT_SMR.1 and FMT_SMF.1	<i>FMT_SMR.1b</i> and <i>FMT_SMF.1b</i>
FMT_MTD.1e	FMT_SMR.1 and FMT_SMF.1	<i>FMT_SMR.1b</i> and <i>FMT_SMF.1b</i>
FMT_REV.1b	FMT_SMR.1	<i>FMT_SMR.1b</i>
FMT_SMF.1b	none	none
FMT_SMR.1b	FIA_UID.1	<i>FIA_UID.2b</i>

FPT_AMT.1	none	none
FPT_RVM.1b	none	none
FPT_SEP.1	none	none
FPT_STM.1b	none	none
ACM_AUT.1	ACM_CAP.3	<u>ACM_CAP.4</u>
ACM_CAP.4	ALC_DVS.1	<u>ALC_DVS.1</u>
ACM_SCP.2	ACM_CAP.3	<u>ACM_CAP.4</u>
ADO_DEL.2	ACM_CAP.3	<u>ACM_CAP.4</u>
ADO_IGS.1	AGD_ADM.1	<u>AGD_ADM.1</u>
ADV_FSP.2	ADV_RCR.1	<u>ADV_RCR.1</u>
ADV_HLD.2	ADV_FSP.1 and ADV_RCR.1	<u>ADV_FSP.2</u> and <u>ADV_RCR.1</u>
ADV_IMP.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1	<u>ADV_LLD.1</u> and <u>ADV_RCR.1</u> and <u>ALC_TAT.1</u>
ADV_LLD.1	ADV_HLD.2 and ADV_RCR.1	<u>ADV_HLD.2</u> and <u>ADV_RCR.1</u>
ADV_RCR.1	none	none
ADV_SPM.1	ADV_FSP.1	<u>ADV_FSP.2</u>
AGD_ADM.1	ADV_FSP.1	<u>ADV_FSP.2</u>
AGD_USR.1	ADV_FSP.1	<u>ADV_FSP.2</u>
ALC_DVS.1	none	none
ALC_FLR.1	none	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	<u>ADV_IMP.1</u>
ATE_COV.2	ADV_FSP.1 and ATE_FUN.1	<u>ADV_FSP.2</u> and <u>ATE_FUN.1</u>
ATE_DPT.1	ADV_HLD.1 and ATE_FUN.1	<u>ADV_HLD.2</u> and <u>ATE_FUN.1</u>
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	<u>ADV_FSP.2</u> and <u>AGD_ADM.1</u> and <u>AGD_USR.1</u> and <u>ATE_FUN.1</u>
AVA_MSU.2	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	<u>ADO_IGS.1</u> and <u>ADV_FSP.2</u> and <u>AGD_ADM.1</u> and <u>AGD_USR.1</u>
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	<u>ADV_FSP.2</u> and <u>ADV_HLD.2</u>
AVA_VLA.2	ADV_FSP.1 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1	<u>ADV_FSP.2</u> and <u>ADV_HLD.2</u> and <u>ADV_IMP.1</u> and <u>ADV_LLD.1</u> and <u>AGD_ADM.1</u> and <u>AGD_USR.1</u>

While the FAU_GEN.2, FIA_UAU.2a, FMT_SMR.1a, FIA_UAU.2b, and FMT_SMR.1b requirements are dependent upon the FIA_UID.1 requirement, the FIA_UID.2 requirement is used in this ST. Note that FIA_UID.2 is hierarchical to FIA_UID.1.

8.5 Explicitly Stated Requirements Rationale

This Security Target contains two explicitly stated requirements: FIA_UAU.2a and FPT_STM.1a. Each of these requirements is based on the CC versions of FIA_UAU.2 and FPT_STM.1, except that these explicitly stated versions specifically allow the IT environment to perform some aspect of the requirement which is not allowed in the original requirements. In the case of FIA_UAU.2a, the authentication function while the TOE enforces restrictions on services until the IT environment confirms the authenticity of applicable users. In the case of FPT_STM.1a, the IT environment 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 IT environment and as such should be considered to satisfy any dependencies levied on FIA_UAU.2 or FPT_STM.1.

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 10 Security Function to TOE SFR Mapping

Security Function	Security Functional Components
Security Audit	FAU_GEN.1a 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
Access Control	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.2a Full residual information protection
	FDP_ROL.1 Basic Rollback
Identification & authentication	FIA_ATD.1a User attribute definition
	FIA_UAU.2a User authentication before any action
	FIA_UID.2a User identification before any action
	FIA_USB.1 User-subject binding
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_REV.1a Revocation
	FMT_SMF.1a Specification of Management Functions
	FMT_SMR.1a Security Management Roles
	TOE Protection
FPT_STM.1a Reliable Time Stamps	

Section 6.2 provides descriptions of how the TOE Security Assurance requirements are satisfied.

8.7 Strength of Function (SOF) Rationale

Although an explicit requirement for the strength of secrets (FIA_SOS.1) is assigned to the IT environment, there are no TOE security functional requirements or security mechanisms that are permutational or probabilistic in nature. Therefore, no strength of function claim is made in this Security Target.