ERSTOE13

Innovation Data Processing

FDRERASE Security Target

Version 1.0

1 July 2005

Prepared for:
Innovation, Inc.

Innovation Plaza, 275 Patterson Avenue
Little Falls, NJ 07424-1658

Prepared By:
Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is FDRERASE provided by Innovation Data Processing, Inc. The TOE is an application run on a mainframe computer with the z/OS or OS/390 operating systems. The purpose of the TOE is to erase data from Direct Access Storage Device Systems (DASD, i.e., a hard disk containing system and user data) that an organization may be scrapping or decommissioning, selling or returning, reusing for a different purpose within the organization or when an organization is leaving a recovery site e.g. after a disaster recovery test, to prevent any access to any data that may reside on the DASD leaving their control. The TOE accomplishes erasure by overwriting DASD, to destroy any data residing on the disk making it no longer accessible. The disk erasure techniques provided by the TOE and described in this Security Target offer successively higher levels of data erasure security by overwriting once or, as appropriate, by overwriting multiple times using multiple data patterns and complements of those patterns, using suitable internal functions to insure data is physically written to disk and to confirm that erasure did take place.

The Security Target contains the following additional sections:

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

1.1 Security Target, TOE and CC Identification

ST Title – Innovation Data Processing, FDRERASE Security Target

ST Version – Version 1.0

ST Date – 1 July 2005

TOE Identification – Innovation Data Processing, FDRERASE, Version 5.4, Level 50.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.2, Revision 256, January 2004.

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.2, Revision 256, January 2004.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.2, Revision 256, January 2004.
    - Part 3 Conformant
    - EAL 2 augmented with ADV_SPM.1 and ALC_FLR.2.

1.3 Conventions, Terminology, Acronyms

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

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Terminology and Acronyms

authorized library	A library containing programs with special privileges.
channel	A functional unit, controlled by the processor, which handles the transfer of data between processor storage and local peripheral equipment. It is part of the processor and contains error recovery logic.
DASD	Direct Access Storage Device – A hard disk.
functional unit	Hardware, software, or a combination of hardware and software that is capable of accomplishing a specified purpose.
ISPF	Interactive System Productivity Facility – An IBM licensed program that provides interactive dialog services.
JCL	Job Control Language – A control language that is used to identify a job to an operating system and to describe the job's requirements.
JES	Job Entry Subsystem – An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs.
logical partition	A subset of a single system that contains resources (processors, memory, I/O devices). A logical partition operates as an independent system. If hardware requirements are met, multiple logical partitions can exist within a system.
LPAR	See 'logical partition'.
SUPERZAP	An IBM-supplied program used to apply binary code fixes ('zaps') to programs.
TSO	Time Sharing Option – Software that provides interactive communications, allowing a user or programmer to start an application from a terminal and work with the application.
vary off/on	To make a device unavailable/available for its normal intended use. A device (such as a DASD) can also be described as being 'varied off' or 'varied on'.

VTOC Volume Table Of Contents – An area on a disk that describes the location, size and other characteristics of each file, library and folder on the disk.

zap IBM terminology for a binary code fix.

In addition, IBM maintains a Web site that contains base computer terminology, terms and definitions as well as the terminology from many of their systems and products in one convenient location. To look up unfamiliar acronyms and terminologies go to:

http://www-306.ibm.com/ibm/terminology/

IBM also provides extensive libraries of manuals on the Web, where additional information on z/OS facilities and programs can be accessed. The IBM z/OS Internet Library can be accessed at:

http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/

2. TOE Description

The Target of Evaluation (TOE) is Innovation Data Processing, FDRERASE, Version 5.4, Level 50.

The TOE is an application that runs on a mainframe computer running the IBM z/OS or OS/390 operating systems. The function of the TOE is to erase data on a Direct Access Storage Device (DASD, i.e., a hard disk). The TOE also offers a capability to verify that user data has been erased.

2.1 TOE Overview

The TOE is an application written in IBM assembler language. The TOE is installed into an authorized library (a library containing programs with special privileges) by a person with update privileges to that library. This person is acting in the role of "TOE Administrator". The TOE is run by a person with execution privileges to that library, as a batch job using IBM JCL (Job Control Language) or the interactive IBM ISPF/TSO facility (similar to a GUI interface). Either method results in the IT environment invoking the TOE and issuing control statements that direct the TOE as to what security function it is to perform. This person is acting in the role of "TOE User".

The TOE provides two different levels of disk erasures. They are the ERASE and SECUREERASE functions. Disk erasures are actually performed by overwriting stored data to make the original data unrecoverable. This overwrite includes the VTOC (Volume Table of Contents) i.e. the disk directory. The TOE also provides a method to verify that user data has been erased. This is the VERIFY function.

The ERASE function overwrites every track of DASD with a track-length record, consisting of binary zeroes by default. This single overwrite will make all data originally on each track unrecoverable by any normal system program running anywhere that has direct access to the disk or through the disk control unit. Original data, however, may still be recoverable through sophisticated laboratory techniques and special programs whose purpose is to recover data on DASD by commanding the disk to skew read heads plus or minus a number of degrees. Any residual data recording on the "edge" of the track may be recoverable using such a technique.

The SECUREERASE function overwrites each DASD track a minimum of three times, writing a random pattern, a complement of the first pattern, and finally another random pattern, by default. This multiple overwrite process (optionally up to eight overwrites) makes the original data unrecoverable, even by sophisticated laboratory techniques applied to hard drives removed from the control unit.

The VERIFY function can be used to sample tracks on the erased volumes to insure that they have been erased. By default it verifies a percentage of the volume but can verify the entire volume if needed.

The TOE runs as an authorized program, from an authorized library. Consequently, the TOE can perform privileged instructions, such as I/O to an offline disk and modification of system control blocks.

The TOE supports three major DASD manufacturers: IBM; EMC; and Hitachi Data. All DASD manufacturers provide a slightly different interface to the disk controller to determine if the overwriting data have actually been written to the DASD. The disk controller command that forces a write is called a "commit." The TOE performs

specific operations to identify the DASD manufacturer at the controller interface so that the appropriate type of commit can be executed to ensure a write has completed.

2.2 TOE Architecture

The TOE architecture is discussed in terms of its physical and logical boundaries. The physical boundaries are visible entities like applications, operating systems, hardware, and software libraries that are part of the TOE or if not part of the TOE environmental entities without which the TOE cannot function. The logical boundaries are the security functions provided by the TOE. In this Security Target they are discussed as the TOE security functions or TSF.

2.2.1 Physical Boundaries

The TOE is an application that is installed into an authorized library (a library containing programs with special privileges). There is no TOE requirement that a "TOE Administrator" and a "TOE User" be separate persons. A person with update (i.e. "TOE Administrator") or execution (i.e. "TOE User") privileges to a library is recognized by the operating system to be operating in one or more of the following roles: trusted systems programmer; storage administrator; operator; or another operating system user granted execution privilege to the library. Any decision to require separation of these roles or not is left to the organizational management authority controlling the TOE site. Any person with execution privileges to the TOE library is hereinafter referred to as the "TOE user".

The intended TOE operating environment is an IBM or IBM compatible mainframe capable of supporting the IBM OS/390 or z/OS operating system, located in a secure environment, i.e. a controlled facility that will prevent unauthorized physical access, where the operating system is securely configured such that it protects the TOE from any unauthorized users or processes and is staffed with trusted, trained and competent individuals.

The z/OS operating system (and its predecessor operating system, OS/390, both collectively referred to hereinafter as z/OS) is the computer operating system for the IBM line of large (mainframe) zSeries servers. IBM zSeries servers provide, among many other features, logical partitions (LPAR) that logically share a computer's clock, processors, memory, and storage so they appear as multiple virtual sets of resources. Each set of resources operates independently with its own operating system instance and applications. Each partition communicates with the other partitions as if the other partition is in a separate independent machine.

The following features of the TOE operating environment (i.e., the z/OS operating system and its underlying hardware) are important to an understanding of the TOE's operation:

- channel programming interface: interface provided by z/OS that enables the TOE to perform direct I/O operations to the disk controller
- disk subsystem: a monolithic unit comprising the disk controller, the disks themselves, and cache memory; it also has error recovery logic
- error recovery software: operating system software that attempts to recover from I/O errors
- console: a display station from which an operator can control and observe the system operation.

Input control statements, interpreted by the TOE to select the functional operation it is to perform, are the external interface to the TOE.

The TOE executes as a batch job under z/OS, which provides these interfaces:

- z/OS JCL, a series of statements that are interpreted by the z/OS Job Entry Subsystem (JES) to define an operating environment, request the execution of the TOE, and specify the location of TOE input control statements and output destination.
- ISPF, an IBM z/OS product providing an interactive end-user interface under z/OS, which can interactively build a batch job (i.e. create the same JCL and control statements as the manual procedure) to execute TOE functions, monitor TOE progress, and suspend or terminate active TOE tasks.
- z/OS Console Command Interface, which allows halt and display commands for the TOE to be input from the operating system console.

The TOE performs direct I/O to the disk controller through the channel programming interface provided by z/OS.

The TOE issues a command (via the channel programming interface) that reserves the DASD exclusively for the operating system instance the TOE is executing on, for all erase operations. This insures that no other system can access the DASD during the erasure. The TOE only overwrites disks that are offline to the operating system. On the operating systems supported by the TOE, untrusted user programs cannot access DASD that is offline. If a DASD is on-line, it would be possible for a user to access the DASD while the TSF was attempting to overwrite the disk. The first check made by the TSF before overwriting a specific DASD is to check to see if the DASD is off-line (optionally, the TOE user can request that the TOE place the DASD offline before starting).. The TOE also makes this same check before each overwrite I/O command during execution. If the TOE finds the DASD is not off-line, the TOE reports this to the TOE user and the TOE will make no attempt to overwrite the data on that specific DASD volume.

Since the TOE is running authorized (privileged), it is allowed to issue these commands to the offline DASD.

The TOE overwrites every track on the DASD in order to erase all prior user and system data, i.e. all data. If this operation fails, the I/O will be automatically retried by the disk subsystem (hardware) and by standard IBM error recovery software in the operating system. If an error is permanent (not recoverable) the TOE reports the error (i.e. a write failure) to the operating system console and to the program listing (TOE user output). The TOE will skip the remainder of the cylinder containing the error and continue overwriting cylinders employing the same algorithm used before the error was encountered until the maximum number of errors is reached. The TOE terminates the erasure after 20 such errors. If any DASD error occurs, even if the maximum error limit is not reached, then when processing for the DASD completes the TSF will terminate with a non-zero completion code (return code) and output an error message with asterisks to the console and program listing, indicating the erasure was incomplete (unsuccessful).

If maintenance is required to fix identified errors in the TOE, Innovation provides them in the form of zaps (binary code fixes) that can be applied with the IBM-supplied SUPERZAP program (only by an authorized user) to the TOE when it is in an inactive mode residing in its authorized library[1].

2.2.2 Logical Boundaries

This section identifies the security functions that FDRERASE provides**.**

2.2.2.1 Security audit

The TOE reports all conditions that indicate a potential failure to successfully complete the disk erasure function (ERASE or SECUREERASE).

The TOE writes to every track on the DASD in order to erase it. If this operation fails, the I/O will be automatically retried by the disk subsystem (hardware) and by standard IBM error recovery software in the operating system. If an error is permanent (not recoverable) the TOE reports the error to the operating system console and to the program listing (TOE user output). If any DASD error occurs, even if the maximum error limit is not reached, the TOE will terminate with a non-zero completion code (return code) and output an error message with asterisks to the console and program listing indicating the erasure was incomplete.

If the TOE finds the DASD is not off-line, the TOE will terminate with a non-zero completion code (return code) and output an error message with asterisks to the console and program listing indicating the erasure was incomplete, and the TOE will make no attempt to overwrite the data on that specific DASD volume.

A completion code may be displayed on the system console or in the TOE program listing, depending on operating system options.

[1] Potential users of the TOE should be aware that application of a zap to the TOE will take the TOE out of its evaluated configuration.

2.2.2.2 User data protection

The TOE provides two disk erasure functions: ERASE and SECUREERASE. Both functions overwrite DASD to ensure the risk of remaining residual data, if any, is commensurate with the risk of a person scavenging for user data. The ERASE function overwrites the DASD with one pass (or more, selectable by an input option, up to 8) of binary zero or of hexadecimal bytes chosen by the TOE user. The SECUREERASE function overwrites a DASD volume with a minimum of three passes (or more, selectable by an input option, up to 8) of hexadecimal bytes determined by the TOE.

In addition, the TSF provides the VERIFY function to enable the TOE user to verify that physical tracks of the DASD have indeed been overwritten sufficiently that no residual information remains.

2.2.2.3 Security management

The TOE provides two disk erasure options and identifies the DASD to be cleared.

The TOE reports to the TOE user the outcome of a DASD overwrite, including: success; failure to access the DASD because the DASD is found to be on-line; and failure to overwrite a bad disk track after successive attempts.

The TOE provides the VERIFY function, to enable the user to verify that physical tracks of a DASD have indeed been overwritten sufficiently that no residual information remains.

2.2.2.4 Protection of the TSF

The TOE protects against failure with loss of the secure state, which requires that the TOE preserve a secure state in the face of the identified failures. The TOE ensures that only DASD that has been varied off-line is available to the TOE. Since the DASD must be off-line, there is no untrusted external interface to the DASD while the TOE is in operation. To ensure this, the first thing the TOE does before execution is to determine if a DASD is off-line. If it is not, the TOE will not attempt to overwrite the DASD and will report the failure to the TOE user. Also, the TOE checks before every write to see if the disk has been varied online; if so, the operation will be terminated with an error message.

The TOE determines the manufacturer of the DASD before beginning to execute. This test is necessary since the external interface of the DASD for committing data to be written from a cache to the hard drive (termed “hardening”) varies by manufacturer, and the TOE has to determine the type and size of DASD it is attempting to overwrite. Before the TOE begins overwriting the DASD, a series of tests is performed to determine the manufacturer and the architecture of the disk to ensure the overwrite occurs appropriately.

Throughout the process of performing a DASD overwrite, the TOE continually monitors for any I/O errors on the write and other I/O issued to the disk. IBM channel hardware and IBM software error recovery is invoked to recover from errors if possible. If all recovery attempts fail, the track is reported to the user and logged for future analysis and the TOE skips the area and continues to overwrite. The user is warned that data may still reside on the damaged area, since it was impossible to write to that area of the DASD because of physical damage. If the hardware will not allow a portion of the DASD to be overwritten, then to absolutely ensure no data is accessible, the failing hard disks may need to be physically removed and destroyed.

During an overwrite of a DASD, if twenty write errors are encountered, the TOE sends a message to the console and the TOE user identifying the DASD, and that the overwrite was a failure. The TOE then terminates: it ceases to exist in an active mode (i.e., resident in memory under operating system control), and automatically returns to its inactive maintenance mode (i.e., resident in the authorized library on disk where it was originally installed).

2.2.2.5 Resource utilization

The TOE notifies the user an operation did not complete in the event of identified failures.

When a failure to write to a specific area of DASD occurs because of damage to the surface of the DASD, the TSF makes multiple attempts to write to the area in an attempt to overwrite any data that may reside there. If this fails, the TOE will skip the affected area and can continue the overwrite until the complete DASD volume has been overwritten.

3. Security Environment

This section summarizes the threats addressed by the TOE and assumptions about the intended environment of the TOE. Note that while the identified threats are mitigated by the security functions implemented in the TOE, the overall assurance level (EAL 2 augmented with ADV_SPM.1 and ALC_FLR.2) also serves as an indicator of whether the TOE would be suitable for a given environment.

3.1 Threats

T.Data_remains_after_clear

Any person with programmatic access to the OS can access data on a DASD through programmatic means after the DASD has been cleared.

T.Data_remnants_remain_after_sanitize

Any person can access data remaining on a DASD after the DASD has been sanitized, through programmatic means or specialized off-line and off-site attempts to recover data from electro-magnetic remnants of recorded data.

T.Data_scavenging

Any person with physical or programmatic access to a DASD that has been overwritten can exploit predictable overwrite patterns to analytically recover data from it.

T.Errant_overwrite

The TOE user invokes the TOE to overwrite a DASD with an inappropriate erase function, thus leaving the data that was on the DASD at an unacceptable risk of compromise.

T.Incomplete_overwrite

Unbeknownst to the TOE user, the TOE fails to completely overwrite a DASD, due to write failures, partial overwrites, or the DASD being on-line and accessible to another program, thus resulting in data remaining on the DASD when the TOE user believes it is completely erased.

3.2 Assumptions

A.Authorized_library

The TOE is installed in an authorized library in the TOE operating environment, such that only appropriately privileged users can install and execute it.

A.Competent_administration

The persons responsible for administration of the TOE environment and installation of the TOE are trusted, trained, competent, and follow all applicable guidance documentation.

A.Competent_use

The persons responsible for execution of the TOE are trusted, trained, competent, and follow all applicable guidance documentation.

A.DASDs_offline

All disks being overwritten are not accessible by user programs.

A.I&A

The TOE operating environment requires users to be identified and authenticated.

A.Secure_environment

The processing resources of the TOE will be located within controlled access facilities that will prevent unauthorized physical access. Furthermore, the underlying operating system operates correctly and is securely configured such that the operating system protects the TOE from any unauthorized users or processes.

A.Security_management

The TOE operating environment supports a security management role and functions to manage its access control policy.

A.Self_protection

The TOE operating environment ensures its own security functions cannot be bypassed, and protects itself from interference and tampering.

A.Proper_procedures

TOE users will abide by all higher authority directives, which could include a second person use of the TOE to verify the person executing the TOE overwrite operation did so on the intended disks, employing appropriate overwrite options.

A.Reliable_clock

The TOE operating environment includes a reliably functioning clock and issues a warning if there is no reliably functioning clock or the clock fails.

4. Security Objectives

This section summarizes the security objectives for the TOE and its environment.

4.1 Security Objectives for the TOE

O.Erase_clears

The TOE shall provide a capability to erase all data on a DASD so as to make the data inaccessible to an operating system user running a non-authorized or authorized program attempting to read the original data, running anywhere that has access to the DASD.

O.Record_of_operation

The TOE shall provide to the user a record of its operation for each overwrite function. Upon completion of an overwrite function, the TOE shall provide a record of the DASD overwritten and the status of the overwrite results for each DASD overwritten. The record will indicate if the overwrite was successful or unsuccessful. Furthermore, the TOE shall report to the user, during the progress of an overwrite function, the following occurrences that indicate the overwrite may not be successful: write failures to the DASD; other I/O failures to the DASD; the DASD coming on-line.

O.Secure_erase_sanitizes

The TOE shall provide a capability to securely erase all data on a DASD so as to make the data inaccessible: to an operating system user running a non-authorized or authorized program attempting to read the original data, running anywhere that has access to the DASD; through the use of off-line special tools designed to perform data recovery or by sophisticated laboratory techniques applied to hard drives removed from the control unit.

O.Verified_operation

When the TSF has completed an erase function, there is significant assurance that the erase function has performed appropriately and overwritten every track on the DASD. The TOE shall provide a capability to verify that DASD have been properly erased and to report to the user the overwrite technique used in the last overwrite.

4.2 Security Objectives for the IT Environment

OE.OS_access_control

The underlying operating system will protect the TOE from unauthorized access and modification.

OE.OS_I&A

The underlying operating system will require operating system users to be identified and authenticated.

OE.OS_reliable_clock

The underlying operating system will provide a reliably functioning clock and will issue a warning if there is no reliably functioning clock or the clock fails.

OE.OS_security_management

The underlying operating system will provide support for a security management role and functions to manage the operating system's access control policy.

OE.OS_self_protection

The underlying operating system will ensure its own security functions cannot be bypassed, and will protect itself from interference and tampering.

4.3 Security Objectives for the Environment

OE.Competent_staff

Those responsible for the TOE will ensure the administrative staff of the TOE environment and the users of the TOE are trusted, trained, competent, and follow all applicable guidance documentation.

OE.Auth_access

Those responsible for the TOE will ensure the TOE operating environment is correctly configured to support the operation of the TOE.

OE.Controlled_facility

Those responsible for the TOE will ensure physical access to the TOE and its operational environment is controlled so that unauthorized physical access to the TOE is prevented.

OE.Operational_procedures

Those responsible for the TOE will ensure the capabilities of the TOE will be utilized in accordance with any higher authority directives or operational procedures.

5. IT Security Requirements

This section defines the security functional requirements for the TOE as well as the security assurance requirements against which the TOE has been evaluated.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_ARP.1: Security alarms
	FAU_GEN.1: Audit data generation
	FAU_SAA.1: Potential violation analysis
FDP: User data protection	FDP_RIP.2: Full residual information protection
	FDP_RIV_EXP.1: Residual information verification
FMT: Security management	FMT_SMF.1a: Specification of Management Functions
FPT: Protection of the TSF	FPT_FLS.1: Failure with preservation of secure state
	FPT_RCV.4 Functional recovery
	FPT_RVM.1a: Non-bypassability of the TSP
FRU: Resource utilization	FRU_FLT.2: Limited fault tolerance

Table 1: TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Security alarms (FAU_ARP.1)

FAU_ARP.1.1 The TSF shall take **[action to display on an operating system console that disk overwrite did not complete. The TSF shall display error messages (with asterisks) on the console and in the audit record]** upon detection of a potential security violation.

5.1.1.2 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions;
b) All auditable events for the **[*not specified*]** level of audit; and
c) **[execution of ERASE, SECURE ERASE and VERIFY function]**.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity, 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,
[for ERASE, SECURE ERASE: write failures, partial overwrites;
for VERIFY: read failures, identification of previously performed overwrite function]

5.1.1.3 Potential violation analysis (FAU_SAA.1)

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:
a) Accumulation or combination of **[write failures, partial overwrites]** known to indicate a potential security violation;
b) **[DASD on-line]**.

5.1.2 User data protection (FDP)

5.1.2.1 Full residual information protection (FDP_RIP.2)

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

Application Note: In the context of this TOE, the objects are the DASDs targeted for erasure and the resources are the tracks to which information content (i.e., data) is written.

5.1.2.2 Residual information verification (FDP_RIV_EXP.1)

FDP_RIV_EXP.1.1 The TSF shall provide the capability to verify that any previous information content of a resource is no longer available on any object in the TSC.

5.1.3 Security management (FMT)

5.1.3.1 Specification of Management Functions (FMT_SMF.1a)

FMT_SMF.1a.1 The TSF shall be capable of performing the following security management functions**:**
[- allow selection of an appropriate overwrite option;
- verification that the data on the DASD has been successfully overwritten].

5.1.4 Protection of the TSF (FPT)

5.1.4.1 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: **[write failures to disk, or abnormal termination]**.

5.1.4.2 Function Recovery (FPT_RCV.4)

FPT_RCV.4.1 The TSF shall ensure that **[the ERASE SF and SECURE ERASE SF, for the following failure scenario: abnormal termination of the SF on the target DASD caused by an accumulation of 20 disk write failures]** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

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

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.

5.1.5 Resource utilization (FRU)

5.1.5.1 Limited fault tolerance (FRU_FLT.2)

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: **[write failures to disk, or abnormal termination]**.

5.2 IT Environment Security Functional Requirements

The following table identifies the SFRs that are assumed to be provided in the environment of the TOE.

Requirement Class	Requirement Component
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
FMT: Security management	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialisation
	FMT_SMF.1b: Specification of management functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1b: Non-bypassability of the TSP
	FPT_SEP.1: TSF domain separation
	FPT_STM.1: Reliable time stamps

Table 2: IT Environment Security Functional Components

5.2.1 User data protection (FDP)

5.2.1.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **[Discretionary Access Control SFP]** on **[subjects: users; objects: libraries, programs; operations: install, execute]**.

5.2.1.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **[Discretionary Access Control SFP]** to objects based on the following: **[subjects: users - privileges; objects: libraries, programs – resource class; operations: install, execute]**.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[The TSF shall allow a user to execute a program if the user has execution privilege to the program's resource class. The TSF shall allow a user to install a program in a library if the user has update privilege to the library's resource class]**.

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

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

5.2.2 Identification and authentication (FIA)

5.2.2.1 User authentication before any action (FIA_UAU.2)

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

5.2.2.2 User identification before any action (FIA_UID.2)

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

5.2.3 Security management (FMT)

5.2.3.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **[Discretionary Access Control SFP]** to restrict the ability to **[*change_default*, *modify*]** the security attributes **[privileges, resource classes]** to **[administrator]**.

5.2.3.2 Static attribute initialisation (FMT_MSA.3)

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

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

5.2.3.3 Specification of Management Functions (FMT_SMF.1b)

FMT_SMF.1b.1 The TSF shall be capable of performing the following security management functions: **[manage user and object security attributes]**.

5.2.3.4 Security management roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[administrator]**.

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

5.2.4 Protection of the TSF (FPT)

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

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

5.2.4.2 TSF domain separation (FPT_SEP.1)

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

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.2.4.3 Reliable time stamps (FPT_STM.1)

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

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ADV_SPM.1 and ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.2: Configuration items
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.1: Descriptive high-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance

Requirement Class	Requirement Component
ALC: Life cycle support	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.1: Developer vulnerability analysis

Table 3: EAL 2 augmented with ADV_SPM.1 and ALC_FLR.2 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Configuration items (ACM_CAP.2)

ACM_CAP.2.1d The developer shall provide a reference for the TOE.
ACM_CAP.2.2d The developer shall use a CM system.
ACM_CAP.2.3d The developer shall provide CM documentation.
ACM_CAP.2.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.2.2c The TOE shall be labeled with its reference.
ACM_CAP.2.3c The CM documentation shall include a configuration list.
ACM_CAP.2.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.2.5c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.2.6c The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.2.7c The CM system shall uniquely identify all configuration items.
ACM_CAP.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)

5.3.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2d The developer shall use the delivery procedures.
ADO_DEL.1.1c 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.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

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)

5.3.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1d The developer shall provide a functional specification.

ADV_FSP.1.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2c The functional specification shall be internally consistent.
ADV_FSP.1.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4c The functional specification shall completely represent the TSF.
ADV_FSP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Descriptive high-level design (ADV_HLD.1)

ADV_HLD.1.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.1.1c The presentation of the high-level design shall be informal.
ADV_HLD.1.2c The high-level design shall be internally consistent.
ADV_HLD.1.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.1.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.1.5c 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.1.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.1.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.1.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal correspondence demonstration (ADV_RCR.1)

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.

5.3.3.4 Informal TOE security policy model (ADV_SPM.1)

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)

5.3.4.1 Administrator guidance (AGD_ADM.1)

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 behavior 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.

5.3.4.2 User guidance (AGD_USR.1)

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 behavior 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)

5.3.5.1 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.2.2c 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.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c 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.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.
ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.
ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.
ALC_FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 Evidence of coverage (ATE_COV.1)

ATE_COV.1.1d The developer shall provide evidence of the test coverage.
ATE_COV.1.1c The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
ATE_COV.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.2 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.
ATE_FUN.1.2d The developer shall provide test documentation.
ATE_FUN.1.1c The test documentation shall consist of test plans, 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.

5.3.6.3 Independent testing - sample (ATE_IND.2)

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

5.3.7.1 Strength of TOE security function evaluation (AVA_SOF.1)

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.

5.3.7.2 Developer vulnerability analysis (AVA_VLA.1)

AVA_VLA.1.1d The developer shall perform a vulnerability analysis.

AVA_VLA.1.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.1.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2c The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3c 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.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security audit

When a failure occurs during a disk overwrite due to disk surface failures, the TSF makes a series of attempts to overwrite a bad disk track; however if the overwrite is unsuccessful, the TSF reports the error to the operating system console and on a TOE job report. The TSF will then skip the remainder of the cylinder (up to 15 tracks) containing the bad track and continue overwriting tracks using the same algorithm used before the bad track was encountered. The TSF reports to the OS console and requests the OS to print a report identifying the track that could not be overwritten and the problem encountered.

The TSF generates an audit report (the TOE job report) for the following auditable events:

- execution of the ERASE function
- execution of the SECURE ERASE function
- execution of the VERIFY function.

The audit report identifies the start and finish of the selected function. Since each invocation of the TOE involves execution of exactly one function, and the audit report is always generated, this is equivalent to auditing the start-up and shutdown of the audit function.

The audit report includes the following information for each auditable event:

- Date and time that event commenced
- Type of event (i.e., ERASE, SECURE ERASE, or VERIFY)
- Subject identity (always FDRERASE)
- Event outcome (success or failure).

The audit report will include, as appropriate for the ERASE and SECURE ERASE functions, any write failures or partial overwrites. The audit report will include, as appropriate for the VERIFY function, any read failures and identification of the overwrite operation most recently performed on the DASD.

If a DASD were to be brought on-line before or during the overwrite, a user recognized by the OS could still have access to a data set that had been granted prior to the commencement of the overwrite and corrupt the overwrite, so the TSF will terminate the erasure, and report the error to the console and the TOE job report. The only access to a DASD varied off-line is through privileged programs. Users are warned of the impact of interrupting the TSF during operation.

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

- FAU_ARP.1: As indicated above, if the TSF detects a potential security violation, it notifies the TOE user by writing a message to an operating system console and to the TOE job report (audit report).
- FAU_GEN.1: As indicated above, the TSF generates a report of the execution of the ERASE, SECURE ERASE, and VERIFY functions, including the date and time the function executed, its outcome, and information relevant to any occurrences of write errors, partial overwrites, read failures, and the most recently performed overwrite operation.
- FAU_SAA.1: The TSF monitors the occurrence of write errors and partial overwrites on the target DASD, which could indicate a potential security violation (i.e., not all data gets overwritten). The TSF also monitors for the target DASD coming on-line during execution of ERASE or SECURE ERASE.

6.1.2 User data protection

The TOE can be instructed through an IBM JCL batch job or interactive IBM ISPF/TSO interface to perform either an ERASE or a SECURE ERASE on one or more specified DASDs. Both ERASE and SECURE ERASE overwrite (i.e., erase) all data on the specified DASDs with default, random or a TOE user selected hexadecimal pattern. After an erase, no residual information remains that can be accessed programmatically in the TOE's intended environment.

When the ERASE function is selected, the TOE user has an option to select the pattern(s) used to overwrite DASD tracks. A pattern is normally a hexadecimal byte, used to overwrite all locations on the disk. If the option 'FE' is selected, the TSF will generate random hexadecimal bytes using an internal proprietary algorithm and overwrite all data on the DASD with the random hexadecimal bytes. From 1 to 8 passes on each track can be selected, and a different pattern can be selected for each pass.

When the SECURE ERASE function is selected, the TSF selects a random hexadecimal byte for the first pass followed by the ones complement of the byte from the first pass to reverse all bits in a hexadecimal byte. A third overwrite is performed with a different random hexadecimal byte. More than 3 overwrites can be selected to meet other requirements; the fourth pattern will be the one's complement of the third pattern If more than 4 passes are requested, passes 5 through 8 will use patterns of random values, instead of using the same value in every byte. Each byte in each record in a cylinder will be randomly generated; this is the same processing as described for the 'FE' option of the ERASE function. The fifth pattern is randomly generated, the sixth pattern will be the complement of the fifth pattern, the seventh pattern will be a new random pattern and the eight will be its complement.

The TOE can also be instructed through an IBM JCL batch job or interactive IBM/TSO interface to perform a VERIFY function. The VERIFY function attempts to read selected areas of a specified DASD to verify that all data previously written on the DASD has actually been overwritten. The TOE user can also specify that the VERIFY function read every track on the specified DASD. The VERIFY function reports the pattern used in the last overwrite of each track it checks.

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

- FDP_RIP.2: The ERASE and SECURE ERASE functions overwrite all data on specified DASDs. Every resource (i.e., track) is effectively deallocated from the objects within the TSC (i.e., the specified DASDs) and any previous information content of every resource is made unavailable by overwriting.
- FDP_RIV_EXP.1: The VERIFY function can be selected by the TOE user to verify that all tracks on a specified DASD have been overwritten by the ERASE or SECURE ERASE function, thereby verifying that any previous information content is no longer available.

6.1.3 Security management

The TOE provides two overwrite operation functions and identifies the DASD to be cleared. If the ERASE function is selected, the TOE user may: choose to allow the TOE to overwrite with binary zeros; select a pattern for the single overwrite; or direct the TSF to generate a random pattern for the single overwrite. The TSF enforces random patterns and their complements for the SECURE ERASE function (3 or more passes).

The TOE provides the VERIFY function to determine that an overwrite function completed. The VERIFY attempts to read selected areas of the DASD to verify that all data previously on the DASD has actually been overwritten.

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

- FMT_SMF.1a: The TSF provides the capability for the user to select the overwrite function appropriate to the user's requirements and specify options to direct the operation of the overwrite function. The TSF also provides the capability for the user to verify that an overwrite function completed successfully.

6.1.4 Protection of the TSF

The TSF protects against failure with loss of the secure state, which requires that the TSF preserve a secure state in the face of the identified failures. The TSF ensures that only DASD that has been placed off-line ('varied off-line' in IBM terminology) is available to the TSF. Since the DASD must be off-line, there is no untrusted external interface to the DASD while the TOE is in operation. To ensure this, the first thing the TSF does before execution is to

determine if a DASD is off-line. If it is not, the TSF will not attempt to overwrite the DASD and will report the failure to the user. Also, the TSF checks before every write to see if the disk has been varied online; if so, the operation will be terminated with an error message.

Since the TSF operates with DASD manufactured by multiple vendors, the TSF must ensure that a commit sent to the Disk Controller actually causes data to be physically written. Forcing a write is typically called a "commit." However some manufacturers do not support the commit command in the same way IBM supports it. To ensure that bytes of data are actually written to disk when the TSF requires this operation, the TSF first tests the DASD to determine the manufacturer. Based on the DASD manufacturer, the TSF selects the appropriate instruction sequence to force an actual write on DASD to occur or to wait for the write to complete. If the commit fails, then some disk data was not hardened to the physical disk so an erase failure is reported to the user, although specific tracks cannot be identified.

Throughout the process of performing a DASD overwrite, the TSF continually monitors for any I/O errors on the write and other I/O issued to the disk. IBM channel hardware and IBM software error recovery is invoked to recover from errors if possible. When I/O commands are issued, I/O errors will indicate if some data could not be written to the DASD. If all recovery attempts fail, the track is reported to the TOE user (via an operating system console) and logged for future analysis and the TSF skips the area and continues to overwrite. The TOE user is warned that user data may still reside on the damaged area, but that it was impossible to write to that area of the DASD because of physical damage. If the hardware will not allow a portion of the DASD to be overwritten, then to absolutely ensure no data is accessible, the failing hard disks may need to be physically removed and destroyed.

During operation of either the ERASE or the SECURE ERASE function on a specified DASD, if twenty write errors are encountered, the TSF sends a message to the console and the TOE user identifying the DASD, and that the overwrite was a failure. The TSF then terminates and must be restarted.

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

- FPT_FLS.1: The secure state of the TSF is that all selected DASDs are completely overwritten, or that the user is made aware of any DASD that was not completely overwritten. The TSF ensures that the selected erase function completes successfully, or notifies the user of any circumstance (DASD write errors, other I/O or environment failures, abnormal termination) that prevented any track on the DASD from being overwritten.
- FPT_RCV.4: The ERASE and SECURE ERASE functions will ensure that every track on every specified DASD is overwritten in accordance with the function options selected by the TOE user. If either function encounters twenty write errors when attempting to overwrite a specified DASD, the function will notify the TOE user that the overwrite has been unsuccessful and will terminate. The TSF enters a consistent and secure state such that it is not executing and the TOE user is made aware that the overwrite function did not complete successfully.
- FPT_RVM.1a: The TSF is invoked via an EXEC directive specifying the security function to be executed (ERASE, SECURE ERASE or VERIFY) and its parameters. There is no other interface to the TSF. The TSF ensures the ERASE and SECURE ERASE functions cannot be bypassed by ensuring it has exclusive access to the target DASD. If the TSF detects that exclusive access cannot be guaranteed (because the DASD is on-line), it terminates with an appropriate message to the TOE user that the overwrite operation did not complete.

6.1.5 Resource utilization

The TOE notifies the user an operation did not complete in the event of identified failures.

When a failure to write to a specific area of DASD occurs because of damage to the surface of the DASD, the TSF makes multiple attempts to write to the area in an attempt to overwrite any user data that may reside there. If this fails the TSF will skip the affected area and continue the overwrite until the complete DASD volume has been overwritten

In the event the TOE terminates abnormally (whether due to errors caused by a failure of a necessary environmental component, such as a disk controller error or operating system failure, or by a user canceling the TOE before an

erase completes), the TSF will not record that the DASD has been erased and once the TSF is restarted it will begin overwriting from the beginning of the DASD.

The Resource utilization function is designed to satisfy the following security functional requirement:

- FRU_FLT.2: The TOE's capability is to completely erase all data from the target DASD, or to notify the user that the erase operation did not complete. If the TSF encounters disk write errors or other circumstances that prevent it completely overwriting the target DASD, it will notify the TOE user accordingly.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by Innovation ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Innovation performs configuration management on the following configuration items:

- TOE code,
- Design information,
- Test documentation,
- User guidance (no administrator role that is separate),
- Delivery and operation information,
- Vulnerability analysis documentation,
- Flaw remediation information (bug tracking),
- Security policy model information,
- and the CM documentation.

Furthermore Innovation performs configuration management of all TSF tests.

These activities are documented in the following Configuration Management (ACM_CAP.2) Configuration Items submission:

- ERSCFM 1.1 - The Innovation Data Processing Software Development Configuration Management Developer Guide

This document identifies Innovation Data Processing uses several facilities to manage change tracking and code protection:

- the Library Management (LM) functions of IBM's ISPF (Interactive System Productivity Facility) are used to manage a hierarchy of product libraries
- IBM's RACF security product is used to limit access to product libraries

The library hierarchy is:

- developer libraries (individual by developer)
- TEST (test versions of product)
- BETA (beta versions of products, which may be shipped to customers willing to test)
- CURP (current production – the libraries containing the current version shipped to customers)
- Version libraries (corresponding to the release number, e.g., V5450)

In each level of the hierarchy there are libraries for

- ASM (source)
- CLIST, MESSAGES, PANELS, SKELETON (ISPF interface components)
- ICL (Installation Control Library – additional documentation and jobstreams)
- JCL (example jobstreams taken from the user documentation)
- LIST (assembly listings of the source modules)
- LOAD (program library of assembled source modules)
- PTF (all non-source (superzap) maintenance applied to or created for a specific version)
- TESTS (all test jobstreams used to verify the version)

The Configuration management assurance measure satisfies the following assurance requirement:

- ACM_CAP.2.

6.2.2 Delivery and operation

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

These activities are documented in the following Delivery and Operation (ADO_DEL.1 and ADO_IGS.1) Delivery, Installation, Generation and Start-up submission:

- ERSDOP 1.0 - INNOVATION Data Processing Software Distribution Process Description and Software Distribution Facility User Guide

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

- ADO_DEL.1
- ADO_IGS.1.

6.2.3 Development

Innovation has documents describing all facets of the design of the TOE. The TSF and its external interfaces are described. The description includes the purpose and method of use of all of the TSF interfaces, and all of the security functions are completely described. Furthermore, the informal TOE security policy model is described.

These documents describe all the TOE interfaces (both external and between subsystems), the high level design of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

The TOE design is documented in the following Development Activity (ADV_FSP.1, ADV-HLD.1, ADV_RCR.1 and ADV_SPM.1) Functional Specification, High-Level Design, Representation Correspondence and Security Policy Model submission:

- ERSDES10 - FDRERASE Solution Functional Specification and High-Level Design Document

The Development assurance measure satisfies the following assurance requirements:

- ADV_FSP.1
- ADV_HLD.1
- ADV_RCR.1
- ADV_SPM.1.

6.2.4 Guidance documents

Innovation provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE. The guidance describes the administrative functions and interfaces available to TOE administrators, any assumptions regarding user behavior that are relevant to the secure operation of the TOE, and all security parameters (including secure values as appropriate) that are under the control of the administrator.

The administrator and user guidance is provided in the following Guidance Documents (AGD_ADM.1 and AGD_USR.1) Administrator and User Guidance submission:

- ERSDOC10 - FDRPAS and FDRERASE User Manual and Installation Guide

The Guidance documents assurance measure satisfies the following assurance requirements:

- AGD_ADM.1
- AGD_USR.1.

6.2.5 Life cycle support

Innovation employs a process where security flaws discovered by customers and Innovation are tracked and corrected by the developer. Innovation bug reconciliation process provides assurance that the TOE is maintained and flaws are corrected in the TOE, first by superzaps to the programs and later by source updates.

Innovation describes the method used to provide flaw information, correction and guidance on corrective actions to TOE users in the following Life Cycle Support (ALC_FLR.2) Flaw Remediation submission:

- ERSBUG 1.1 - Innovation Data Processing Software Product Life Cycle Maintenance Support (Bug Track) User Guide

The Life cycle support assurance measure satisfies the following assurance requirement:

- ALC_FLR.2.

6.2.6 Tests

The test documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

These activities are documented in the following Test Activity (ATE_COV.1, ATE_FUN.1 and ATE_IND.2) Coverage, Functional and Independent Testing submission:

- ERSTST 1.0 - Innovation Data Processing Testing Procedures and FDRERASE Test Documentation

The Tests assurance measure satisfies the following assurance requirements:

- ATE_COV.1
- ATE_FUN.1
- ATE_IND.2 – to be accomplished by the evaluation team.

6.2.7 Vulnerability assessment

Innovation has conducted a strength of function analysis to identify and analyze permutational or probabilistic security mechanisms implemented in the TOE. The ST does not specify any security functional requirements or security functions that involve permutational or probabilistic mechanisms. Therefore, an overall strength of function claim is not applicable, and the assurance requirements of AVA_SOF.1 are vacuously satisfied.

Innovation performs regular vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE. Innovation's Vulnerability analysis describes the obvious vulnerabilities identified in the TOE, the operating system, and the various DASD manufactures on which the TSF operate. Each candidate vulnerability that might apply to the TSF is reviewed to determine if there is a vulnerability that can be exploited in the TSF. For each vulnerability that could be exploited in the TSF, the vulnerability is treated as a bug requiring immediate remedy and work begins immediately to remedy the problem. Customers are informed of the vulnerability as described in the Life Cycle Support (ALC_FLR.2) Flaw Remediation submission; ERSBUG 1.1 - Innovation Data Processing Software Product Life Cycle Maintenance Support (Bug Track) User Guide, using the INNOVATION "News via Email" service and additionally, in the case of a security vulnerability as a further precaution, the "contact" at all effected customer sites is sent a notice by US Postal Service regular mail and a patch is released and installed as soon as it is available.

These activities are documented in the following Vulnerability Assessment (AVA_VLA.1) submission:

- ERSVUL 1.0 - Innovation Data Processing FDRERASE Vulnerability Assessment

The Vulnerability assessment assurance measure satisfies the following assurance requirements:

- AVA_SOF.1
- AVA_VLA.1.

7. Protection Profile Claims

There is no Protection Profile claim in this Security Target.

8. Rationale

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

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

8.1 Security Objectives Rationale

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

	T.Data_remains_after_clear	T.Data_remnants_remain_after_sanitize	T.Data_scavenging	T.Errant_overwrite	T.Incomplete_overwrite
O.Erase_clears	X				
O.Record_of_operation					X
O.Secure_erase_sanitizes		X	X		
O.Verified_operation				X	
OE.Operational_procedures				X	

Table 4: Threats to Objectives Correspondence

8.1.1 T.Data_remains_after_clear

Any person with programmatic access to the OS can access data on a DASD through programmatic means after the DASD has been cleared.

This threat is countered by the TOE security objective O.Erase_clears. This objective will ensure the TOE erases all data on the target DASD, rather than just logically removing the data (e.g., by deleting VTOC entries).

8.1.2 T.Data_remnants_remain_after_sanitize

Any person can access data remaining on a DASD after the DASD has been sanitized, through programmatic means or specialized off-line and off-site attempts to recover data from electro-magnetic remnants of recorded data.

This threat is countered by the TOE security objective O.Secure_erase_sanitizes. This objective will ensure the TOE securely erases all data on the target DASD such that it cannot be: read through programmatic means; retrieved using off-line or sophisticated laboratory techniques applied to hard drives removed from the control unit.

8.1.3 T.Data_scavenging

Any person with physical or programmatic access to a DASD that has been overwritten can exploit predictable overwrite patterns to analytically recover data from it.

This threat is countered by the TOE security objective O.Secure_erase_sanitizes. This objective will ensure the TOE securely erases all data on the target DASD such that it cannot be: read through programmatic means; retrieved using off-line or sophisticated laboratory techniques applied to hard drives removed from the control unit.

8.1.4 T.Errant_overwrite

The TOE user invokes the TOE to overwrite a DASD with an inappropriate erase function, thus leaving the data that was on the DASD at an unacceptable risk of compromise.

This threat is countered by the TOE security objective O.Verified_operation, supported by the environment security objective OE.Operational_procedures. The objective O.Verified_operation ensures the TOE can be invoked to verify that a target DASD has been completely overwritten and to report to the TOE user the overwrite technique used on the final overwrite pass. This will provide the TOE user with the information necessary to determine if the appropriate erase function has been applied to the target DASD. The objective OE.Operational_procedures supports O.Verified_operation, since the verify function is not automatic and local procedures may be required to ensure an erased DASD is subsequently verified.

8.1.5 T.Incomplete_overwrite

Unbeknownst to the TOE user, the TOE fails to completely overwrite a DASD, due to write failures, partial overwrites, or the DASD being on-line and accessible to another program, thus resulting in data remaining on the DASD when the TOE user believes it is completely erased.

This threat is countered by the TOE security objective O.Record_of_operation. This objective will ensure that the TOE generates a record of the result of each erase function performed on each DASD. The report will identify if the erase function completed successfully (i.e., no data remains on the DASD), or, if it did not complete successfully, the reason the function did not complete. The TOE user will therefore be able to determine the result of all requested erase functions.

	A.Authorized_library	A.Competent_administration	A.Competent_use	A.DASDs_offline	A.I&A	A.Secure_environment	A.Security_management	A.Self_protection	A.Proper_procedures	A.Reliable_clock
OE.Auth_access	X					X				
OE.Competent_staff	X	X	X	X		X				
OE.Controlled_facility						X				
OE.Operational_procedures									X	
OE.OS_access_control	X									
OE.OS_I&A					X					
OE.OS_security_management							X			
OE.OS_self_protection								X		
OE.OS_reliable_clock										X

Table 5: Assumptions to Objectives Correspondence

8.1.6 A.Authorized_library

The TOE is installed in an authorized library in the TOE operating environment, such that only appropriately privileged users can install and execute it.

This assumption is satisfied by the environment security objectives OE.OS_access_control, OE.Competent_staff and OE.Auth_access. The objective OE.OS_access_control ensures the underlying operating system provides appropriate mechanisms to protect the TOE. The objective OE.Competent_staff, by ensuring the TOE environment administrative staff follows applicable guidance, also ensures the TOE is correctly and securely installed according to the instructions in the guidance documentation. The objective OE.Auth_access ensures the TOE operating environment is configured correctly to support the operation of the TOE, including configuration of the authorized library in which the TOE is installed.

8.1.7 A.Competent_administration

The persons responsible for administration of the TOE environment and installation of the TOE are trusted, trained, competent, and follow all applicable guidance documentation.

This assumption is satisfied by the environment security objective OE.Competent_staff. This objective will ensure the administrative staff of the TOE environment are trusted, trained, competent, and follow all applicable guidance.

8.1.8 A.Competent_use

The persons responsible for execution of the TOE are trusted, trained, competent, and follow all applicable guidance documentation.

This assumption is satisfied by the environment security objective OE.Competent_staff. This objective will ensure the users of the TOE are trusted, trained, competent, and follow all applicable guidance.

8.1.9 A.DASDs_offline

All disks being overwritten are not accessible by user programs.

This assumption is satisfied by the environment security objective OE.Competent_staff. This objective will ensure the administrative staff of the TOE environment and the users of the TOE will follow all applicable guidance. The guidance documentation for the TOE provides clear instructions to ensure all disks to be overwritten are offline on all systems, and therefore not accessible by user programs.

8.1.10 A.I&A

The TOE operating environment requires users to be identified and authenticated.

This assumption is satisfied by the environment security objective OE.OS_I&A. This objective will ensure the underlying operating system identifies and authenticates its users.

8.1.11 A.Secure_environment

The processing resources of the TOE will be located within controlled access facilities that will prevent unauthorized physical access. Furthermore, the underlying operating system operates correctly and is securely configured such that the operating system protects the TOE from any unauthorized users or processes.

This assumption is satisfied by the environment security objectives OE.Auth_access, OE.Competent_staff and OE.Controlled_facility. The objective OE.Auth_access ensures the TOE operating environment is configured correctly to support the operation of the TOE. The objective OE.Competent_staff, by ensuring the TOE environment administrative staff follows applicable guidance, also ensures the TOE environment is correctly and securely configured according to the instructions in the guidance documentation. The objective OE.Controlled_facility ensures the TOE and its operational environment is protected from unauthorized physical access.

8.1.12 A.Security_management

The TOE operating environment supports a security management role and functions to manage its access control policy.

This assumption is satisfied by the environment security objective OE.OS_security_management. This objective will ensure the underlying operating system supports a security management role and functions to manage its access control policy.

8.1.13 A.Self_protection

The TOE operating environment ensures its own security functions cannot be bypassed, and protects itself from interference and tampering.

This assumption is satisfied by the environment security objective OE.OS_self_protection. This objective will ensure the underlying operating system does not allow its security functions to be bypassed and protects itself from interference and tampering.

8.1.14 A.Proper_procedures

TOE users will abide by all higher authority directives, which could include a second person use of the TOE to verify the person executing the TOE overwrite operation did so on the intended disks, employing appropriate overwrite options.

This assumption is satisfied by the environment security objective OE.Operational_procedures. This objective will ensure the capabilities of the TOE, including the capability to verify an erase function, will be utilized in accordance with any higher authority directives or operational procedures.

8.1.15 A.Reliable_clock

The TOE operating environment includes a reliably functioning clock and issues a warning if there is no reliably functioning clock or the clock fails.

This assumption is satisfied by the environment security objective OE.OS_reliable_clock. This objective will ensure the underlying operating system provides a reliably functioning clock and issues a warning if there is no reliably functioning clock or the clock fails.

8.2 Security Functional Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 6** indicates the requirements that effectively satisfy the individual objectives.

	O.Erase_clears	O.Record_of_operation	O.Secure_erase_sanitizes	O.Verified_operation	OE.OS_access_control	OE.OS_I&A	OE.OS_security_management	OE.OS_self_protection	OE.OS_reliable_clock
FAU_ARP.1		X							
FAU_GEN.1		X		X					
FAU_SAA.1		X							
FDP_RIP.2	X		X						
FDP_RIV_EXP.1				X					
FMT_SMF.1a	X		X	X					
FPT_FLS.1	X		X						
FPT_RCV.4	X		X						
FPT_RVM.1a	X		X	X					
FRU_FLT.2	X		X						
FDP_ACC.1					X				
FDP_ACF.1					X				
FIA_UAU.2						X			
FIA_UID.2						X			
FMT_MSA.1							X		
FMT_MSA.3							X		
FMT_SMF.1b							X		
FMT_SMR.1							X		
FPT_RVM.1b								X	
FPT_SEP.1								X	
FPT_STM.1									X

Table 6: Objectives to Requirements Correspondence

8.2.1 O.Erase_clears

The TOE shall provide a capability to erase all data on a DASD so as to make the data inaccessible to an operating system user running a non-authorized or authorized program attempting to read the original data, running anywhere that has access to the DASD.

This TOE security objective is satisfied by the TOE security functional requirement FDP_RIP.2, supported by FMT_SMF.1a, FPT_FLS.1, FPT_RCV.4, FPT_RVM.1a, and FRU_FLT.2.

FDP_RIP.2 specifies the requirement that all information content of a resource (i.e., a track) will be made unavailable when it is deallocated from an object (i.e., a DASD). This requirement provides the capability to erase all data on a DASD. FMT_SMF.1a provides the capability to select the appropriate overwrite operation to erase the data. FPT_RCV.4 ensures that the ERASE security function will either complete successfully or, in the event the TSF encounters 20 write failures and terminates abnormally, will recover to a consistent and secure state. The secure state is that either all data is erased, or the erase function reports upon termination that the erase did not complete. In addition, FRU_FLT.2 and FPT_FLS.1 ensure the maintenance of a secure state in the event of write failures to disk, or abnormal termination. FPT_RVM.1a ensures the erase function is invoked and cannot be bypassed.

8.2.2 O.Record_of_operation

The TOE shall provide to the user a record of its operation for each overwrite function. Upon completion of an overwrite function, the TOE shall provide a record of the DASD overwritten and the status of the overwrite results for each DASD overwritten. The record will indicate if the overwrite was successful or unsuccessful. Furthermore, the TOE shall report to the user, during the progress of an overwrite function, the following occurrences that indicate the overwrite may not be successful: write failures to the DASD; other I/O failures to the DASD; the DASD coming on-line.

This TOE security objective is satisfied by the TOE security functional requirements FAU_ARP.1, FAU_GEN.1, and FAU_SAA.1.

FAU_GEN.1 specifies the requirement for generation of an audit record of the execution of the erase functions, indicating the success or failure of the functions. FAU_SAA.1 specifies the requirement to monitor the occurrence of write failures, errant overwrites, partial overwrites, and the target DASD going on-line. Each of these events represents a potential security violation. FAU_ARP.1 specifies the requirement to write a message to the operating system console and the audit record upon detection of a potential security violation (i.e., write and other I/O failures to the DASD, or the DASD going on-line).

8.2.3 O.Secure_erase_sanitizes

The TOE shall provide a capability to securely erase all data on a DASD so as to make the data inaccessible: to an operating system user running a non-authorized or authorized program attempting to read the original data, running anywhere that has access to the DASD; through the use of off-line special tools designed to perform data recovery or by sophisticated laboratory techniques applied to hard drives removed from the control unit.

This TOE security objective is satisfied by the TOE security functional requirement FDP_RIP.2, supported by FMT_SMF.1a, FPT_FLS.1, FPT_RCV.4, FPT_RVM.1a, and FRU_FLT.2.

FDP_RIP.2 specifies the requirement that all information content of a resource (i.e., a track) will be made unavailable when it is deallocated from an object (i.e., a DASD). This requirement provides the capability to securely erase all data on a DASD. FMT_SMF.1a provides the capability to select the appropriate overwrite operation to erase the data such that it is not recoverable even by sophisticated laboratory techniques. FPT_RCV.4 ensures that the ERASE security function will either complete successfully or, in the event the TSF encounters 20 write failures and terminates abnormally, will recover to a consistent and secure state. The secure state is that either all data is erased, or the erase function reports upon termination that the erase did not complete. In addition, FRU_FLT.2 and FPT_FLS.1 ensure the maintenance of a secure state in the event of write failures to disk, or abnormal termination. FPT_RVM.1a ensures the erase function is invoked and cannot be bypassed.

8.2.4 O.Verified_operation

When the TSF has completed an erase function, there is significant assurance that the erase function has performed appropriately and overwritten every track on the DASD. The TOE shall provide a capability to verify that DASD have been properly erased and to report to the user the overwrite technique used in the last overwrite.

This TOE security objective is satisfied by the TOE security functional requirements FDP_RIV_EXP.1 and FAU_GEN.1, supported by FMT_SMF.1a, and FPT_RVM.1a.

FDP_RIV_EXP.1 specifies the requirement for a capability to verify that any previous information content of a resource is no longer available on any object in the TOE Scope of Control. That is to say, that every track of the specified DASD has been overwritten with the appropriate pattern to make the data unrecoverable to the desired extent (dependent on the perceived risk of compromise). FAU_GEN.1 specifies the requirement for generation of an audit record of the execution of the verify function, indicating the success or failure of the function. FMT_SMF.1a specifies the requirement to provide the capability to verify that data on the DASD has been successfully overwritten. FPT_RVM.1a ensures the verify function is invoked and cannot be bypassed.

8.2.5 OE.OS_access_control

The underlying operating system will protect the TOE from unauthorized access and modification.

This security objective for the IT environment is satisfied by the IT environment security functional requirements FDP_ACC.1 and FDP_ACF.1. These requirements work together to define and specify a Discretionary Access Control (DAC) policy that provides controls on who can install and execute the TOE.

8.2.6 OE.OS_I&A

The underlying operating system will require operating system users to be identified and authenticated.

This security objective for the IT environment is satisfied by the IT environment security functional requirements FIA_UAU.2 and FIA_UID.2, which ensure users accessing the IT Environment of the TOE have been successfully identified and authenticated.

8.2.7 OE.OS_reliable_clock

The underlying operating system will provide a reliably functioning clock and will issue a warning if there is no reliably functioning clock or the clock fails.

This security objective for the IT environment is satisfied by the IT environment security functional requirement FPT_STM.1, which specifies the requirement to provide reliable time stamps. The time stamps will be deemed reliable if they are generated from a reliably functioning clock, and if the IT environment issues a warning if there is no reliably functioning clock or if the clock fails.

8.2.8 OE.OS_security_management

The underlying operating system will provide support for a security management role and functions to manage the operating system's access control policy.

This security objective for the IT environment is satisfied by the IT environment security functional requirements FMT_MSA.1, FMT_MSA.3, FMT_SMF.1b and FMT_SMR.1.

FMT_SMR.1 specifies that the IT environment supports an administrator as a security management role. FMT_SMF.1b specifies a capability to manage the user and object security attributes that control the operation of the DAC policy, while FMT_MSA.1 and FMT_MSA.3 specify how those security attributes can be managed, and that management is restricted to the administrator.

8.2.9 OE.OS_self_protection

The underlying operating system will ensure its own security functions cannot be bypassed, and will protect itself from interference and tampering.

This security objective for the IT environment is satisfied by the IT environment security functional requirements FPT_RVM.1b and FPT_SEP.1.

FPT_SEP.1 specifies the requirement that the IT Environment of the TOE maintains a security domain for its own execution and enforces separation between the security domains of its subjects. This ensures the IT Environment protects the TOE from tampering while it is executing, while FPT_RVM.1b ensures all of the security policy enforcement functions in the IT environment are invoked and succeed before other functions can proceed.

8.3 Security Assurance Requirements Rationale

The target assurance level is EAL2, augmented with ADV_SPM.1 and ALC_FLR.2.

EAL2 was selected as the base assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. FDRERASE is targeted at a relatively benign environment with good physical access security and competent TOE administrators and users. Within such environments, it is assumed that attackers will have a low attack potential. As such, EAL2 is appropriate to provide the assurance necessary to counter the limited potential for attack.

ALC_FLR.2 is selected as an appropriate augmentation because flaw remediation procedures provide greater assurance that security-related bugs will be fixed in a widely distributed commercial product.

ADV_SPM.1 is selected as an augmentation partly because it is a dependency of two of the security functional requirements claimed for the TOE. However, it is also viewed as desirable to provide a clear statement of the TSP that is also evaluated for correspondence with the functional specification.

8.4 Strength of Functions Rationale

The TOE does not specify any security functional requirements or implement any security functions that involve permutational or probabilistic mechanisms. Therefore, an overall strength of function claim is not applicable.

8.5 Requirement Dependency Rationale

The following table demonstrates that all dependencies among the claimed security requirements are satisfied and therefore the requirements work together to accomplish the overall objectives defined for the TOE.

ST Requirement	CC Dependencies	ST Dependencies
FAU_ARP.1	FAU_SAA.1	FAU_SAA.1
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_SAA.1	FAU_GEN.1	FAU_GEN.1
FDP_RIP.2	none	none
FDP_RIV_EXP.1	none	FDP_RIP.2
FMT_SMF.1a	none	none
FPT_FLS.1	ADV_SPM.1	ADV_SPM.1
FPT_RCV.4	ADV_SPM.1	ADV_SPM.1
FPT_RVM.1a	none	none
FRU_FLT.2	FPT_FLS.1	FPT_FLS.1
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1, FMT_MSA.3
FIA_UAU.2	FIA_UID.1	FIA_UID.2 (hierarchical)
FIA_UID.2	none	none
FMT_MSA.1	(FDP_ACC.1 or FDP_IFC.1), FMT_SMF.1, FMT_SMR.1	FDP_ACC.1, FMT_SMF.1b, FMT_SMR.1

ST Requirement	CC Dependencies	ST Dependencies
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	FMT_MSA.1, FMT_SMR.1
FMT_SMF.1b	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.2 (hierarchical)
FPT_RVM.1b	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none
ACM_CAP.2	none	none
ADO_DEL.1	none	none
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1	ADV_RCR.1
ADV_HLD.1	ADV_FSP.1, ADV_RCR.1	ADV_FSP.1, ADV_RCR.1
ADV_RCR.1	none	none
ADV_SPM.1	ADV_FSP.1	ADV_FSP.1
AGD_ADM.1	ADV_FSP.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1	ADV_FSP.1
ALC_FLR.2	none	none
ATE_COV.1	ADV_FSP.1, ATE_FUN.1	ADV_FSP.1, ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1
AVA_SOF.1	ADV_FSP.1, ADV_HLD.1	ADV_FSP.1, ADV_HLD.1
AVA_VLA.1	ADV_FSP.1, ADV_HLD.1, AGD_ADM.1, AGD_USR.1	ADV_FSP.1, ADV_HLD.1, AGD_ADM.1, AGD_USR.1

Table 7: TOE Security Requirement Dependencies

8.6 Explicitly Stated Requirements Rationale

This Security Target specifies an explicitly stated requirement: FDP_RIV_EXP.1.

FDP_RIV_EXP.1 specifies a unique requirement for the product type of the TOE that is not covered by any security functional requirement defined in Part 2 of the Common Criteria. The TOE type is a data erasure product, and the TOE is intended for a specific environment (IBM OS/390 and z/OS operating systems) and specific data storage objects (IBM, EMC, and Hitachi Data DASDs). The CC provides the Residual Information Protection (FDP_RIP) family within the User Data Protection (FDP) class of security functional requirements to address the need to ensure that deleted information is no longer accessible. However, the CC does not provide a requirement to verify that information that should have been made inaccessible has in fact been made inaccessible. One objective of the TOE is to provide the user appropriate assurance that the data that was on a DASD has been made inaccessible, commensurate with the perceived level of risk. For example, the user may wish to dispose of the DASD and wants to be confident that all data previously stored on the DASD has been erased sufficiently that its recovery is impractical if the DASD should end up "in the wrong hands". FDP_RIV_EXP.1 has been specified to address the need for such a verification capability in the TOE product type.

FDP_RIV_EXP.1 clearly has a dependency on FDP_RIP.2, since it would be meaningless to attempt to verify that previous information content had been made unavailable if there was no requirement to ensure previous information content had been made unavailable. This dependency is identified in Table 6.

8.7 TOE Summary Specification Rationale

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

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 8: Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	User data protection	Security management	Protection of the TSF	Resource utilization
FAU_ARP.1	X				
FAU_GEN.1	X				
FAU_SAA.1	X				
FDP_RIP.2		X			
FDP_RIV_EXP.1		X			
FMT_SMF.1a			X		
FPT_FLS.1				X	
FPT_RCV.4				X	
FPT_RVM.1a				X	
FRU_FLT.2					X

Table 8: Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.