HP Archive Backup System for OpenVMS Version 4.5 Release Notes

December 2008

This release note describes the enhancement, known restrictions, and errors found in the Archive Backup System software and documentation, and provides a brief description of the changes in this version.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface

Conventions

The following conventions are used in this document:

Convention	Meaning
{}	In format command descriptions, braces indicate required elements.
[ ]	In format command descriptions, square brackets indicate optional elements of the command syntax. You can omit these elements if you wish to use the default responses.
boldface text	Boldface type in text indicates the first instance of a term defined in the Glossary or defined in text.
italic text	Italic type emphasizes important information, indicates variables, indicates complete titles of manuals, and indicates parameters for system information.
Starting test ...	This type font denotes system response, user input, and examples.
Ctrl/*x*	Hold down the key labels Ctrl (Control) and the specified key simultaneously (such as Ctrl/Z).
PF1 *x*	The key sequence PF1 x instructs you to press and release the PF1 key, and then press and release another key (indicated here by x).
n	A lowercase italic n denotes the generic use of a number. For example, 19nn indicates a four-digit number which the last two digits are unknown.
x	A lowercase x denotes the generic use of a letter. For example, xxx indicates any combination of three alphabetic characters.

Related Products

The following related products may be mentioned in this document:

Product	Description
HSM	HSM refers to Hierarchical Storage Management for OpenVMS software.
MDMS	MDMS refers to Media, Device and Management Services for OpenVMS software.
OpenVMS	OpenVMS refers to OpenVMS operating system.
SLS	SLS refers to Storage Library System for OpenVMS software.

Associated Documents

The following documents are part of the Archive Backup System documentation set:

- *HP Archive Backup System for OpenVMS Installation Guide*
- *HP Archive Backup System for OpenVMS Guide to Operations*
- *HP Archive Backup System for OpenVMS Command Reference Guide*
- *Quick Reference Guide to Configure ABS/MDMS*

  The *Quick Reference Guide to Configure ABS/MDMS* is intended for all the novice users of HP Archive Backup System for OpenVMS.

1 Introduction

This document describes the enhancement, known problems, restrictions, and errors found in the Archive Backup System (ABS) software and documentation.

- ABS Version 4.5 uses the Media, Management and Device Services (MDMS) server to manage policies (saves, restores, archives, and environments). For more information on using the new database, see the MDMS sections in the release notes.

Note

Before upgrading from V3.x, HP recommends that you perform an ABS SHOW object/FULL/ OUT=filename on all your ABS objects. This is useful for comparison of the objects after the upgrade and the conversion.

- This release of ABS/MDMS includes ABS$DATA_COLLECTION.COM command procedure. An utility used to gather the system related information. This reduces the initial interaction effort to gather the system related information from you and the Field Engineers when a problem is filed.

  This command procedure is located in the ABS$ROOT:[SYSTEM] directory.

- During the upgrade to Version 4.5 from V3.x, the MDMS database files are copied to files named MDMS$DATABASE_LOCATION:MDMS$*_ DB.DAT_COPY. The ABS policy engine configuration file is also renamed to ABS$SYSTEM:ABS$POLICY_CONFIG.DAT_OLD and the ABS$SYSTEM:ABS$EXT_*.COM files are renamed to ABS$EXT_*.COM_ OLD. These files are used if you remove the Version 4.5 files. For more information on removing Version 4.5 files, see "How to Remove Version 4.5 ABS/MDMS Files".

- ABS Version 4.0 did NOT allow a rolling upgrade on the servers or clients. This is because the policy engine has been moved to the MDMS server, so all nodes must use the same mechanism to talk to the database server. If you are on an ABS V4.x stream, you can use the rolling upgrade option.

- The MDMS Version 4.5 server provides the scheduling of save and restore requests. For more information on the scheduling options, see the MDMS sections of the release notes.

- If you are upgrading from V3.x, the DECscheduler scheduling option will be converted to SCHEDULER during the installation. Jobs will be submitted to DECscheduler using the MDMS$SYSTEM:MDMS$EXT_SCHEDULER.COM command procedure.

Note

Because of this change, the SCHEDULER option will not work if you do not have a license for DECscheduler. If you are using the DECscheduler for flexibility of scheduling, check the options available in Version 4.5 for the INTERNAL schedule with custom schedules. It offers very flexible scheduling.

- The ABS$SYSTEM:ABS$EXT*.COM external scheduling files were changed between V3.x and V4.0. In V3.x, the files were called ABS$SYSTEM:ABS$EXT_*.COM. In V4.0, these were moved and renamed to MDMS$SYSTEM:MDMS$EXT*.COM. If you are upgrading from V3.x and if you have made modifications to these procedures, ensure that you incorporate your modifications into the new versions. Take the changes from ABS$SYSTEM:ABS$EXT_*.COM and put them to the MDMS$SYSTEM:MDMS$EXT*.COM files. The installation will not incorporate your changes into the command procedures, it will provide a new command procedure. The new command procedure must be used for the Version 4.5 external scheduling to work.

- ABS Version 4.5 does not require DECnet for server communications. Becasue the MDMS server handles the communication, you may use TCP/IP and/or DECnet.

- If you are using the MDMS Graphical User Interface (GUI) on Microsoft Windows and if Java is installed, then typing **java** in the command line should execute **java.exe** from the installed location. However, if this does not happen, you will need to first edit the **MDMSView.bat** file to include the correct path of **java.exe**.

  Example: C:\Program Files\Java\j2re1.4.2_06\bin\java.exe

- When using the MDMS GUI on OpenVMS Alpha, the account quota PGFLQUO should be set to at least 200,000. Correspondingly, the system pagefile should be large enough to accommodate this quota. The process quota WSQUOTA should be set to 10,000. Correspondingly, the sysgen parameter WSMAX, should be set to 10,000 or higher. The system where you are running the GUI should have at least 128 MB of memory (preferably higher).

- When you install ABS and and use Hierarchical Storage Management (HSM) on your system, the installation procedure prompts you to use the backup-via-shelving paradigm. If you answer "yes", a logical will be defined in your ABS$SYSTARTUP.COM file. If you say "no", the logical is not defined.

  If you later decide to use the backup-via-shelving paradigm, you must define the logical ABS$SHELVING_SUPPORT to TRUE in your ABS$SYSTARTUP.COM procedure. Failure to define the logical will mean that the /noshelved and /nopreshelved qualifiers will not be used during backup and files will be unshelved during the backup.

- Earlier versions of ABS had allowed the use of SLS/MDMS V2.9x as the media manager; this interface no longer exists in ABS. You must be prepared to use MDMS Version 4.5 before upgrading. For information and guidelines on MDMS V2.9x to MDMS Version 4.5 conversion, see the chapter, "Converting to ABS/MDMS V4.X" cX" in the HP Archive Backup System for OpenVMS, Guide to Operations.

- ABS Version 4.5 does not support RDB database for ABS policies. If you are currently using ABS with an RDB database you must convert the RDB database using the conversion procedure described in the release notes for V3.2A before upgrading to ABS Version 4.5.

1.1 Installing ABS/MDMS

If you are installing ABS/MDMS for the first time, see the *HP Archive/Backup System for OpenVMS Installation Guide* before installing the products. For information on configuring ABS/MDMS, see the HP Archive Backup System Guide to Operations.

The ABS/MDMS Version 4.5 kit contains the following savesets:

- ABS045.A - Installation procedure and Release Note
- ABS045.B - Common files
- ABS045.C - OpenVMS Alpha Executables
- ABS045.D - RDF kit
- ABS045.E - GUI kit for OpenVMS Alpha/Integrity servers
- ABS045.F - GUI kit for Windows
- ABS045.G - OpenVMS Integrity server Executables

_____ **Note** _____

The VAX saveset will no longer be provided with the ABS/MDMS kit; you will find only seven savesets.

_____

2 Software Version Requirements

The following sections describe the software version requirements for ABS and MDMS.

2.1 Archive Backup System Software Version Requirements

Ensure that the following requirements are met before installing ABS Version 4.5:

- OpenVMS Alpha V7.3-2, V8.2 and V8.3
- OpenVMS Integrity servers V8.2-1, V8.3, and V8.3-1H1
- HP Media Robot Utility (MRU), V1.8B is provided with ABS/MDMS Version 4.5. Install this software if you plan to use the direct robotic control features.
- DECnet Phase IV or DECnet-Plus for OpenVMS is installed, if you plan to use DECnet for communication.
- TCP/IP Services for OpenVMS is installed, if you plan to support UNIX or Windows clients, or use TCP/IP for server communication.
- Java Runtime Environment (JRE) V1.5 is installed. Installing JRE is optional; it is only required if you plan to use the MDMS GUI.
- OpenSSL is installed to enable the Encryption Key Management feature.

_____ **Note** _____

ABS for OpenVMS works with HP TCP/IP services for OpenVMS. Our limited testing of Multinet with ABS/MDMS has not uncovered any issues. Engineering will not reject an escalation because of the use with this product. However, if, through investigation Multinet is suspected

of being the problem, the customer will be asked to pursue the remedy through their existing contract with Multinet service.

2.2 MDMS Software Version Requirements

MDMS Version 4.5 is supported on the following OpenVMS versions:

- OpenVMS Alpha V7.3-2, V8.2, and V8.3

- OpenVMS Integrity servers V8.2-1, V8.3, and V8.3-1H1

- HP Media Robot Utility (MRU), V1.8B is provided with ABS/MDMS Version 4.5. Install this software if you plan to use the direct robotic control features

- DECnet Phase IV or DECnet-Plus for OpenVMS is installed, if you plan to use DECnet for communication.

- TCP/IP Services for OpenVMS is installed, if you plan to support UNIX or Windows clients, or use TCP/IP for server communication.

- Java Runtime Environment (JRE) V1.5 is installed. Installing JRE is optional; it is only required if you plan to use the MDMS GUI.

- OpenSSL to enable the Encryption Key Management feature

In addition, the GUI is supported on the following OpenVMS versions for MDMS Version 4.5:

- OpenVMS Alpha V7.3-2, V8.2, and V8.3

- OpenVMS Integrity servers V8.2-1, V8.3, and V8.3-1H1

- Windows Intel Platforms (2000/XP)

The GUI requires the Java Runtime Environment (JRE) V1.5 or Java Development Kit (JDK) V1.5. In addition, the GUI communicates with the MDMS server on an OpenVMS system using the TCP/IP protocol. Therefore, a product supporting the TCP/IP protocol (example: TCP/IP Services for OpenVMS or Multinet) is a pre-requisite for using the MDMS GUI.

If you are using ABS Version 4.5 for the first time, MDMS Version 4.5 can be installed and configured as a new installation.

- For OpenVMS Alpha servers, you can upgrade from any ABS V3.x, or V4.x environments to ABS Version 4.5.

- For OpenVMS Integrity servers you can upgrade from any ABS V4.3, V4.3A, V4.4, V4.4A, or V4.4B environments to ABS Version 4.5.

Alternatively, if you have a previous installation of Storage Library System (SLS) /MDMS V2.9x, the SLS to ABS Migration tool provided with the ABS Version 4.5 kit facilitates you to migrate from SLS/MDMS V2.9x to ABS/MDMS V4.x. The migration happens in the following phases:

- Converting MDMS V2.x symbols and databases to MDMS V4.x database objects: To convert the MDMS V2.x TAPESTART.COM, volume and magazine databases into MDMS V4.x database objects, use the following command procedure:

```
$ @MDMS$SYSTEM:MDMS$CONVERT_V2_TO_V4
```

- Applying Prev3 Support: Set the MDMS$PREV3_SUPPORT logical to "TRUE" in SYS$MANAGER:MDMS$STARTUP.COM to restore the SLS backed up data after the conversion.

- Converting SLS SBK Symbols to ABS Policy objects: To convert the SLS SBK Symbols to ABS Policy objects, use the following command procedure:

  ```
  $ @ABS$SYSTEM:SLS_CONVERT.COM
  ```

3 New Features/Enhancements

The following section describes the new features and enhancement included in ABS/MDMS Version 4.5.

3.1 Data Encryption in ABS/MDMS

ABS/MDMS Version 4.5 is enhanced to support data encryption, thereby securing data files. This feature is an extension of the Encryption feature available in the OpenVMS Backup utility. All the supported features, restrictions, and methods to use the encryption feature in Backup are made available in ABS/MDMS. Because Backup utility enables you to backup and encrypt the OpenVMS data files, the encryption feature in ABS is limited to OpenVMS data files only.

Note

Currently the feature of cataloging the existing savesets is not supported for Encrypted savesets.

For Disaster Recovery (DR) backups, encryption must not be enabled.

3.1.1 Overview of OpenVMS Backup Utility

The OpenVMS Backup utility enables you to take Selectives, Images, and Incremental Backups of data files and helps to restore these objects. It also provides other additional features, such as Encryption and so forth.

The BACKUP/ENCRYPT command is used to create encrypted savesets. To create an encrypted saveset, specify the /ENCRYPT qualifier anywhere on the BACKUP command line. You can also specify either a key name or a key value, but not both. By default, the algorithm selected is DESCBC. BACKUP uses these user specified keys and algorithms to encrypt data. If you have already defined the key value using /ENCRYPT=NAME=key name, you can specify the key name directly in the Backup command as shown. Execute the following command to create backup with encryption:

```
$BACKUP/ENCRYPT= ([key_name] [, ALGORITHM=algorithm])
```

To define a key value, specify /ENCRYPT=VAUE=key-value, where key-value can be created using a combination of one of the following characters:

- 1 to 243 alphanumeric characters enclosed in quotation marks (""), dollar sign ($), and underscore. The key is not case-sensitive.

- A hexadecimal constant using the digits 0 to 9 and A to F.

Note

For additional security, specify the /ENCRYPT qualifier without any parameters and press "Return". The command prompts you for a key

value. When you enter the value, the software does not display the value you have typed, and for verification purpose, the software prompts you to retype the value.

OpenVMS Backup supports the following encryption algorithms:

- DESCBC
- DESCFB
- DESECB
- AESCBC128
- AESCBC192
- AESCBC256
- AESECB128
- AESECB192
- AESECB256
- AESCFB128
- AESCFB192
- AESCFB256
- AESOFB128
- AESOFB192
- AESOFB256

The ENCRYPT utility is supported on OpenVMS V8.3 and later versions. For systems running on OpenVMS 8.2, install the Encrypt utility. This does not require a separate license.

For systems running on OpenVMS 7.3-2, install the ENCRYPT utility with a valid license for the ENCRYPT utility. Contact your HP Support to purchase the Encrypt utility.

OpenVMS Backup utility does not provide any key management functionality. As the number of keys used to encrypt the data files increases, it becomes difficult to trace or remember the keys. Hence, ABS is enhanced with the key management functionality to automatically generate keys for encrypting data. The keys are stored in a key store file and this file is required while restoring the data files.

3.1.2 Enabling the Encryption Feature

To enable the Encryption feature, upgrade earlier versions of ABS to ABS Version 4.5 or perform a complete installation of ABS Version 4.5. The key management functionality is used to encrypt data. However, if you do not require the key management functionality, follow the procedure described in the section, 3.1.3 "Encrypting Data without the Key Management Functionality".

The key management facility is implemented using a command procedure, a set of images, and a key store file. The key store file will store the keys used for encrypting data, which will be queried later for restoring the data files.

The key store file, ABS$ENCRYPTION_<node name>.DAT, is created in the ABS$SYSTEM directory. This file is created during upgradation or installation of ABS on the system by the FDL file. Note that <node name> is the node where the key store file is stored during encryption.

Apart from the FDL file, the following .EXE files are also provided in the SYSCOMMON:[SYSEXE] for saving and restoring the key:

- ABS$ENCRYPT_SAVE.EXE

- ABS$ENCRYPT_RESTORE.EXE

3.1.3 Encrypting Data without the Key Management Functionality

You can encrypt data without using the key management functionality. Data files can be encrypted by specifying the Agent_Qualifier field in the selection object.

A standard template is used by ABS to format the Backup command. To append any specific Backup qualifiers, you can use the Agent_Qualifier field in the selection object. Any additional qualifiers specified here, will be appended to the template while formatting the Backup command.

The following example illustrates how to specify the /ENCRYPT qualifier with the key name and algorithm in the selection object:

```
        Selection: SINGLE
      Description:
  Access Control: NONE
          Owner: OPAL::TEST
Agent Qualifiers: /ENCRYPT=(Key=ABS_KEY, ALGORITHM=ALGO_NAME)
      Before Date: NONE
Conflict Options: RETAIN_VERSION
Data Select Type: VMS_FILES
         Date Type: MODIFIED
          Exclude:
          Include: LOGIN.COM;*
       Since Date: NONE
      Source Node:
```

where:

- "ABS_KEY" is the key name that must be specified for encrypting the data file. The key must be created using the Encrypt utility and later assigned to a key_name by executing the following command:

  ```
  $ENCRYPT/CREATE/SYS ABS_KEY "value"
  ```

  where:

  "value" can be one of the following and the key is not case-sensitive:

  - 1 to 243 alphanumeric characters enclosed in quotation marks (""), dollar sign ($), and underscore.

  - A hexadecimal constant using the digits 0 to 9 and A to F.

- ALGO_NAME is the algorithm used from the list of supported algorithms on ABS/MDMS.

For more information about using the /ENCRYPT qualifier, see Section 3.1.1, "Overview of the Backup Utility".

Note

The selection must be associated with the save or restore object while saving and restoring data. The key and algorithm used while saving data files must be specified while restoring data files, without which the data will not be decrypted.

3.1.4 Encrypting data with Key Management Functionality

Encrypting data with the key management functionality requires OpenSSL. The OpenSSL generates random numbers; these random numbers are later used to generate keys for taking backup of data files. Hence, to enable encryption in the system, OpenSSL must be installed on all the systems where the ABS saves will be executed.

OpenSSL kits for OpenVMS Integrity servers and OpenVMS Alpha are available at the following web address:

`http://h71000.www7.hp.com/openvms/products/ssl/ssl_download.html`

If OpenSSL is already installed, ensure that the "SSL* logicals are defined by executing the following command:

`$ SHOW LOGICAL *SSL*`

If the logicals related to OpenSSL are not defined, then start OpenSSL by executing the following command procedure:

`$ SYS$COMMON:[SYS$STARTUP]SSL$STARTUP.COM`

Securing the Key Store File

By default, ABS$ENCRYPTION_<node_name>.DAT, the key store file is created in the ABS$ROOT:[SYSTEM] directory. ABS$ENCRYPTION_<node_name>.DAT stores all the keys used for encrypting the data. For security reasons and to prevent unauthorized users from accessing the key store file, HP recommends you to define the ACLs using the OpenVMS DCL command. While defining the ACLs, ensure that ABS is provided with access to the key store file.

Note

If the file is lost due to some reason, all the backed up data is lost. It is, therefore, advised that you secure and protect the file.

The key store file, ABS$ENCRYPTION_<node_name>.DAT is local to the node where the save operation is executed. Therefore, data can be restored only on the node where the key store file is located, else you must copy the key store file to the required node or the server for restoring the data. For more information on restoring the data, see Restoring Data on a Different Node.

For example, if save is executed on node A, it can be restored only on node A and not on other nodes. If there is a requirement to restore from other nodes in the environment, the key store file must be copied to the required system or the server.

3.1.5 Backup/Restoring Data Files with Encryption

This section describes how to backup/restore data with encryption. Complete the following steps to encrypt and decrypt data files:

1. **Setting the Prologue in the Save object**

   To enable encryption while backing up data, the ABS$ENCRYPT_SAVE.EXE must be appended to the save prologue and executed. The ABS$ENCRYPT_SAVE.EXE file generates the required key for encrypting data and also stores the key used for encrypting data in the key store file. Edit the prologue associated with the save object and append the following command:

   ```
   $ RUN SYS$COMMON:[SYSEXE]ABS$ENCRYPT_SAVE.EXE
   ```

   Or

   Create a command procedure using the `$ RUN SYS$COMMON:[SYSEXE]ABS$ENCRYPT_SAVE.EXE` and associate the command procedure to the Save object as its prologue.

2. **Defining the Logical for Encryption Algorithm**

   Define a logical to specify the algorithm for encrypting data.

   For example:

   ```
   $DEFINE/SYSTEM ABS_MDMS_ENCRY_ALGO "DESCBC"
   ```

   The logical, ABS_MDMS_ENCRY_ALGO, is defined to specify the DESCBC algorithm to encrypt data. The algorithms supported in the Backup utility are also supported in ABS Version 4.5.

   ───────────────── **Note** ─────────────────

   If the logical for the algorithm used is not defined, the save operation fails. The algorithm used is stored in the key store file, you need not define the symbol to restore the data file.

   ──

3. **Executing the Saves**

   While executing the saves, in the Selection Object, specify the key name and the algorithm for the Agent_Qualifier field, as shown in the following example:

   ```
    Selection: SINGLE
         Description:
      Access Control:
               Owner:
   Agent Qualifiers:/ENCRYPT=(name=ABS_MDMS_KEY, algo='ABS_MDMS_ENCRY_ALGO_TYPE)
         Before Date: NONE
    Conflict Options: RETAIN_VERSION
    Data Select Type: VMS_FILES
           Date Type: MODIFIED
             Exclude:
             Include:
          Since Date: NONE
         Source Node:
   ```

   where:

   - ABS_MDMS_KEY is the logical defined by SYS$COMMON:[SYSEXE]ABS$ENCRYPT_SAVE.EXE.

- ABS_MDMS_ENCRY_ALGO_TYPE is the symbol defined by the ABS encryption image and will be substituted with the algorithm while executing the save.

The logical name and the symbol name must not be changed in the Selection object.

4. **Setting the Environment Prologue to Restore Object**

To restore the data files from the saved location, you will need the keys used for encrypting the data. The key store file,which is an RMS indexed file, contains the following details about the key that is generated:

- Key name used for encryption/decryption

- Tape label if the backup destination is TAPE

- Location of the save set, if it is a DISK backup

- Saveset name. ABS generates the saveset name with a time stamp, that is unique to the node

- Execution node name, which is the execution node where the backup was executed

- Algorithm used for taking Backup

While restoring the encrypted data, execute ABS$ENCRYPT_RESTORE.EXE in the restore prologue. This image identifies the saveset that must be restored and will retrieve all information, such as the key used for encryption, volumes used for the saveset, and algorithm used for encryption before the restore operation. The ABS$ENCRYPT_RESTORE.EXE image updates the selection object with the required details.

While restoring the saves, in the Selection Object, specify the key name and the algorithm for the Agent_Qualifier field, as shown in the following example:

```
        Selection: SINGLE
      Description:
  Access Control:
            Owner:
 Agent Qualifiers:/ENCRYPT=(name=ABS_MDMS_KEY, algo='ABS_MDMS_ENCRY_ALGO_TYPE)
      Before Date: NONE
 Conflict Options: RETAIN_VERSION
 Data Select Type: VMS_FILES
        Date Type: MODIFIED
          Exclude:
          Include:
       Since Date: NONE
      Source Node:
```

Edit the prologue associated with the restore object and then append the following command:

`$ RUN SYS$COMMON:[SYSEXE]ABS$ENCRYPT_RESTORE.EXE`

Or

Create a command procedure with `$ RUN SYS$COMMON:[SYSEXE]ABS$ENCRYPT_RESTORE.EXE` and associate the same to the restore object as its prologue.

5. **Restoring Data on a Different Node**

   The ABS$ENCRYPTION_<node_name> is a node specific file, all the information pertaining to the savesets that are created by executing backup on this node will be stored in this key store file. During restoration, the ABS$ENCRYPT_RESTORE.EXE will search all the key store files present in the ABS$SYSTEM directory to obtain the information about keys used while taking backup.

   Hence if restoration of data has to be performed on any other node, the key store file corresponding to that node where the data backup was executed has to be copied onto the node where restoration will be executed. Execute the following steps to restore data on a node where you will be performing the restore operation:

   1. Copy the node specific file, ABS$ENCRYPTION_<node_name> to the ABS$SYSTEM directory on the execution node.

   2. If the backups taken from multiple clients are to be restored on one node, all the key store files from the corresponding node to the other clients have to be copied to the restoration node.

   3. Note that you also need to copy the ABS catalogs to the node where restores will be executed if this node does not have access to the catalogs stored on the nodes where backups were executed.

   _____ **Note** _____

   During uninstallation of ABS Version 4.5, the encryption related files are not deleted from the system.

   _____

3.2 ABS is Qualified with Oracle 10gR2

ABS Version 4.5 is qualified with Oracle 10gR2 on OpenVMS Alpha and Integrity Servers. For more information about System Backup to Tape (SBT) for Oracle databases, see "Documentation Updates and Errata".

4 Problems Fixed in this Release

The following section describes the problems fixed in ABS Version 4.5:

4.1 ABS Save Fails with ABS_TLE_NOT_FOUND

Problem:

While running the save operation, if the system time changes because of the DayLight Saving Time (DST) setting, the ABS save operation fails with the "ABS_TLE_NOT_FOUND" error.

Although the expiration or retention date is specified in the archive object, the SCRATCH date of the volume object gets set to NONE after executing the save operation.

To enable the DST feature, define the system wide logical, ABS$DST_CHANGE before the DST setting comes into effect. This will make ABS to effectively address the time shift during ABS operations. Deassign the logical after the time change comes into effect.

To define the logical, execute the following command:

```
$ DEFINE/SYS ABS$DST_CHANGE TRUE
```

To deassign the logical, execute the following command:

```
$ DEASSIGN/SYS ABS$DST_CHANGE
```

Solution:

These problems were reported as QXCM1000390238 and QXCM1000785707, and these problems are fixed in this release.

4.2 RDEV_WRITE_GROUP_DELAY and RDEV_WRITE_GROUP_SIZE Logicals

Problem:

When ABS used the RDF devices, the CPU was consumed upto 78%. Redefining the RDF logicals, RDEV_WRITE_GROUP_DELAY and RDEV_WRITE_GROUP_SIZE did not the improve the performace.

Solution:

This problem was reported as QXCM1000425375 and QXCM1000821968 and is fixed in this release. With this fix, after defining the RDF logicals, RDEV_WRITE_GROUP_DELAY and RDEV_WRITE_GROUP_SIZE, the perfomance has improved by 50% of the earlier value.

Reboot the system to update to the new RDF images after installing ABS Version 4.5.

4.3 ABS/MDMS-GUI Password Handling

Problem:

ABS/MDMS-GUI password handling is not compatible with the UAF /pwdmix flag.

Solution:

This problem was reported as QXCM1000775487, and is fixed in this release.

4.4 Enhanced Support for DECnet over IP

Problem:

In a client-sever environment, while configuring DECnet over IP, which uses only IP address instead of DECnet address, the SYS$NODE_FULLNAME logical is not supported. SYS$NODE_FULLNAME will always have the node_name prefixed with "LOCAL:."

Solution:

This problem was reported as QXCM1000794394, and is fixed in this release. The SYS$NODE logical now contains the node name without the prefix "LOCAL:.". Also, the mail box size has been increased to 512 blocks.

4.5 Using the EXTERNAL scheduler to cleanup MDMS ONE_TIME_ONLY job

Problem:

While using the EXTERNAL scheduler, the cleanup of MDMS ONE_TIME_ONLY job deletes the active batch jobs from the queue.

Solution:

This problem was reported as QXCM1000801979, and is fixed in this release.

5 Known Problems and Restrictions

The following sections describe the known problems and restrictions with ABS and MDMS.

5.1 Archive Backup System Known Problems

The following section describes the known problems that exist from the earlier releases of ABS:

5.1.1 Restriction with SBT

The System Backup to Tape (SBT) feature of ABS for Oracle Recovery Manager (RMAN) does not support the "proxy copy" feature and the DUPLICATE command of ORACLE RMAN.

5.1.2 Volume Placement shows ONSITE after the Volume Database is Converted back to SLS/MDMS V2.x Environment

Reconvertingthe volume database from an ABS/MDMS V4.x environment to an SLS/MDMS V2.x environment, when you provide the STORAGE REPORT VOLUME command from the client node, the volume placement for some of the volumes is shown as OUTSIDE or ONSITE.

This error on volume placement to ONSITE or OUTSIDE location is not consistent across the volumes in the volume database. It is also occasionally observed when you migrate from an SLS/MDMS V2.x to an ABS/MDMS V4.x environment.

A possible workaround to this issue could to take a fresh inventory of volumes after the migration from/to ABS/MDMS V4.x and SLS/MDMS V2.x environments.

5.1.3 Aborted Save Logs Multiple Messages on OpenVMS Integrity Servers

When an ABS save/restore request is aborted, the status "Job Terminated" is written in the Log file multiple times.

5.1.4 SBT Test Program with RDF Devices

SBT is provided as a shareable image, which is invoked by the Oracle Recovery Manager (RMAN) to backup Oracle databases. SBT allocates and mounts the devices for RMAN to backup databases. The SBT test program which is supplied with SBT, is used to test the SBT interface with ABS/MDMS 4.x. An attempt to backup and restore to an RDF device using the SBT test program fails with the following footprint:

```
MMAPI error: err_code=7501, Fatal media movement error
Failed to allocate drive
System Error: %SYSTEM-F-ACCVIO, access violation, reason mask=!XB, virtual
address=!XH, PC=!XH, PS=!XL
%SYSTEM-F-ACCVIO, access violation, reason mask=!XB, virtual address=!XH,
PC=!XH, PS=!XL
```

The problem is seen only with Rdserver running on OpenVMS (Alpha) 6.2-1 and 7.2-1 and Rdclient running on OpenVMS Alpha (7.2-1). The problem is not seen with Rdserver running on OpenVMS (VAX).The problem is seen only when both backup and restore operation is done in a single session from the SBT test program. We are in touch with RDF Engineering to resolve this problem.

However you can still use the RDF with the Rdserver running on OpenVMS (Alpha and VAX) to backup and restore your Oracle database without any problems as long as backup and restore operations of the database are done in separate sessions.

Before configuring SBT, ensure that Oracle's Recovery Manager, is setup and you are able to access it. For more information on configuring SBT, see *System Backup to tape for Oracle Databases* in the ABS Guide to Operations Manual.

5.1.5 One Movement Type Allowed Per Save Request

Only one movement type is allowed per save request. The movement types are FULL, INCREMENTAL, and SELECTIVE. If you specify include specifications for more than one type of movement, you may get backup errors. For example, if your save request contains include specifications with a disk name and also other objects with more than a disk name (that is, disk:[dir]), backup will return an error on some objects because an image backup works with a disk name only.

Workaround:

Include only one type of include specification in your save request. For example, a save request should only include disk names, or directory names.

5.1.6 ABS REPORT SAVE_LOG Fails when Searching SLS Catalogs

If you perform an ABS REPORT SAVE_LOG command and do not specify a catalog or storage_class, once ABS has searched an SLS catalog, the command will stop and will not search the remaining catalogs on the system.

Workaround:

Until this problem has been resolved, specify a catalog or storage_class name on the ABS REPORT SAVE_LOG command.

5.1.7 ABS might have Problems in Saving Files Greater than 2 GB in Size

You might have problems saving files greater than 2 GB in size on UNIX and Windows NT systems. On UNIX systems, see the documentation provided with the gtar you are using. The gtar version that we supply for Windows NT and the source for UNIX do not accommodate files greater than 2 GB in size.

5.1.8 Setting Volume on Archive while Save is Running

Adding a volume set name to an archive while a save request is running can cause problems with the cleanup process after the save request (leaving it in a locked state). Two archives referencing the same volume set is not supported.

5.1.9 Inherited Problems in Backup Agent ABS$BACKUP.EXE

Problem:

If you are using Backup-Via-Shelving, the default backup agent for backing up OpenVMS files is SYS$SYSTEM:ABS$BACKUP.EXE. This image was built using the V6.1 source code of OpenVMS Backup (BACKUP.EXE) with some HSM features added. Thus, ABS$BACKUP has inherited the problems known to exist in V6.1 of BACKUP and the new features added to BACKUP since V6.1 are not available in ABS$BACKUP.

Solution:

The HSM features will be incorporated into BACKUP in an OpenVMS remedial ECO scheduled for release sometime after OpenVMS V7.3.

If you are not using Backup-Via-Shelving, ABS will use SYS$SYSTEM:BACKUP.EXE.

5.1.10 Verifying Installation

Problem:

When installing ABS as a new installation, the database initialization programs might fail to run. In this case, IVP fails with errors failing to show the storage classes and execution environments.

Solution:

After the installation, run SYS$SYSTEM:ABS$DB_INIT.EXE. This executable initializes the database with the default storage classes and execution environments.

5.1.11 Permission Denied or Invalid Argument Error on Windows Save Request

Problem:

ABS may produce the following log file errors after executing an NT save request:

```
ABSgtar: can't add file C:\AFILE.EXT: : Permission denied
ABSgtar: can't open directory C:\ADIR: : Invalid argument
```

The cause of these errors is either because the directory or file is open for write access by a user or application, or the system has been denied read access to the file or directory. ABS runs under the SYSTEM account.

Workaround:

Close all open files or set the access on the files for the SYSTEM account.

To set the file access:

1. Select the file from a **Fileview** window.

2. Select *Properties* from the *File* menu.

3. Click the *Security* tab and then select *Permissions*.

4. Select *Add* and highlight *SYSTEM*.

5. Add the type of access (full control is best, so you can restore files) and click the *Add* button. This provides the SYSTEM account access to the files.

Solution:

This problem will be corrected in the future version of ABS.

5.1.12 Save Request must Specify the Correct Object Type

Problem:

If you create a save request for a Tru64 UNIX or Windows file and you do not specify the UNIX_FILES_GTAR or WINDOWS_NT_FILES_GTAR object type (default is VMS_FILES), ABS produces a scheduler error.

Solution:

Ensure to specify the UNIX_FILES_GTAR or WINDOWS_NT_FILES_GTAR object type for include_specification.

5.1.13 ABS RESTORE with Brief Listing Fails to Provide Listing File

Problem:

If you are executing a RESTORE request which uses an environment policy with a BRIEF_LISTING setting, the restore will not create a listing file.

Workaround:

You may set the environment policy to FULL_LISTING and get a full listing instead.

Solution:

This problem will be addressed in a future version of ABS.

5.1.14 RMU Backup Exclude_Specification Parameter

Problem:

Save Request

If you create or modify a save request for an Oracle RDB storage area and enter an exclude_specification parameter, ABS generates an incorrect syntax for the RMU Backup Utility. As a result, the save request fails.

Solution:

This problem will be corrected in a future version of ABS software.

Problem:

Restore Request

If you create or modify a restore request for an Oracle RDB database or storage area and enter an exclude_specification parameter, ABS ignores the file name entered for exclude_specification parameter.

Solution:

None. The RMU Backup Utility does not support an exclude_specification parameter for a restore request.

5.1.15 RMU Backup Include_Specification Parameter

Problem:

Wildcard Character

The RMU Backup Utility does not support wildcard characters (* or %) in the include_specification parameter for a save or restore request. However, if you enter one of these characters, ABS fails to invalidate the illegal entry. ABS executes the save or restore request but generates an incorrect file name. As a result, the save or restore request fails.

Solution:

For an Oracle RDB database or storage area, do not enter any wildcard characters in the include_specification parameter.

Problem:

Comma-Separated List

If the include_specification parameter on a restore request for an Oracle RDB database or storage area consists of a comma-separated list of disks or file names, ABS may not correctly restore the data.

Example: The include-specification contains:

```
DISK$USER1:[USER1]SITE_PERSONNEL.RDB,DISK$USER2:[USER2]PERSONNEL.RDB
```

Possible result:

If the restore of DISK$USER1:[USER1]SITE_PERSONNEL.RDB includes any incremental restore operations, ABS may terminate the restore operation without restoring the subsequent Oracle RDB file names specified in the list.

Solution:

For each Oracle RDB database or storage area that you want to restore, create a separate restore request or use Add Object option on the GUI to individually add each file name. For more information about how to use the Add Object option, see Table 9-3 in the Archive Backup System for OpenVMS Guide to Operations.

5.1.16 Selective Restore Operation

Problem:

If a selective restore request is performed for an Oracle RDB storage area, and the last backup operation of the Oracle RDB database was an incremental, the restore request fails and ABS generates the following message:

```
%RMU-F-WRNGDBBTYP, save_set_name does not contain a full backup
```

Solution:

Create an incremental restore request so that ABS will restore both full and incremental backups of the storage area.

5.1.17 Avoid Use of ABS$UBS_VERBOSE Logical Name

Problem:

The system logical name ABS$UBS_VERBOSE should be defined only when requested by HP. It may cause adverse side-effects after Save request failures, which include unintended initialization of the output volume. Check to ensure that this logical name has not been defined on nodes running ABS.

5.2 Archive Backup System Restrictions

The following are known restrictions that exist from earlier versions of ABS:

5.2.1 Parallel Save and Restore Requests must not be Executed during Catalog Cleanup

Parallel Save or Restore requests must not be executed when the cleanup of the Volume_Set type of catalog is in progress. If executed, it results in catalog having dangling AOEI entries.

5.2.2 SCSNODE must be Defined for ABS/MDMS to Start

The Sysgen parameter SCSNODE must be defined on your system to run ABS /MDMS. The current startup procedures require this value or the SYS$NODE logical name to be defined.

5.2.3 Using DECnet-Plus (Phase V) with ABS

If you use DECnet-Plus with ABS, you must define Phase IV synonyms (6 characters or less) for your nodes. ABS cannot use the long node names.

5.2.4 Post-processing Command Fails to Execute

Problem:

If a Save request does not complete successfully, and a post-processing command is specified either on the save request or the environment policy, the post-processing command will not execute.

Solution:

None. ABS only executes the post-processing command on the basis of a successful completion. The next time the save request completes successfully, ABS will execute the post-processing command.

5.2.5 Restriction On The "-bnn" Value for UNIX or NT Clients

Problem:

When performing Windows or UNIX save or restore requests, you may have set the agent qualifier "-bnn" to set the block length of a disk. If you set the value of "-bnn" to a number higher than 127 (-b127), you may get unexpected results for the save request. In this situation, ABS may generate ASSERT and ACCVIO errors.

Solution:

This is a restriction in OpenVMS. Do not set the -bnn agent qualifier value to a number higher than 127. If the disk is large enough to exceed this amount, create more than one save request for that particular disk.

More Information:

For more information about the -bnn qualifier, see *Archive Backup System for OpenVMS Guide to Operations*.

5.2.6 Restoring an Entire Tru64 UNIX Filesystem

You cannot restore an entire Tru64 UNIX filesystem to either the original location or to a different location by specifying only a "/" for the include specification. To restore the entire UNIX filesystem, you must specify each top level directory as shown in the following example:

```
/usr
/dev
```

5.2.7 Tape Device Protection

Problem:

When performing a user backup, if the tape device is protected against the user, the following error is logged in the execution log file:

```
THREAD #1:%BACKUP-F-PARSE, error opening
NODE01MKA500:[000000]13JUL19951339120.;
THREAD #1: -RMS-E-PRV, insufficient privilege or file protection violation
```

Workaround:

If this error occurs, check the device protection of the target tape device by entering the following command from a privileged account:

```
$ SHOW SECURITY/CLASS=device
_Object name: MKA500

_MKA500: object of class DEVICE
Owner: [SYSTEM]
Protection: (System: RWPL, Owner: RWPL, Group: RWPL, World)
Access Control List: <EMPTY>
```

To set the protection of the tape device for the world access, enter the following command:

```
$ SET SECURITY/CLASS=DEVICE/PROT=(W:RWPL) MKA500
$ SHOW SECURITY /CLASS=DEVICE MKA500

_MKA500: object of class DEVICE
Owner: [SYSTEM]
Protection: (System: RWPL, Owner: RWPL, Group: RWPL, World: RWPL)
Access Control List: <EMPTY>
```

When the tape drive enables access to the user, the insufficient privilege problem from the backup command is eliminated.

5.2.8 OPCOM Restriction

If you select OPCOM as the method of notification, then you must select the BRIEF option. If you select VERBOSE or NORMAL, the message could be truncated.

5.2.9 Request Restart

ABS does not support restarting a save or restore request in the middle of the operation. All save and restore requests are restarted from the beginning.

5.2.10 No Restart Operation

ABS does not attempt to replace a faulty member of a volume set and, as a result, the save operation fails. ABS also may not take advantage of the restart capabilities built into some of the backup agents.

5.2.11 Restoring Multi-File Oracle RDB Databases

Do not specify an output location while restoring a multi-file Oracle RDB database. If an output location is specified, ABS places all storage area files in the specified output location.

5.2.12 Restrictions with ABS Windows 2000 Client

- Encrypted files not supported

- Backup of open files not supported

- Mount point save and restore operations are successful when mount point exists while restoring. But if the mount point is removed, ABS Windows Client would restore the data of the mounted volumes to a new folder. This folder is no longer a mount point.

- Restoration of Hard Link files results in the hard link being broken and two different individual files being restored.

- The backup agent (ABSgtar) for Windows client does not allow wildcards in the file specification strings for save requests.

5.3 Known MDMS Problems and Restrictions

This section describes known problems and restrictions that exist from the earlier versions of MDMS.

5.3.1 Logical Pointing to Disk Name must be Defined during the Restore Operation

If a logical is used to specify the disk name, which is part of the destination specified for the restore request (the location where the data is restored), ensure that the logical is defined before executing the restore request.

If the destination is not exclusively provided for the /DESTINATION qualifier, the restore request considers the include specification path as the default destination. If the logical is not properly defined, the restore request fails displaying the NO_SUCH_DEVICE error.

5.3.2 Specifying Wildcard * to Maintain Directory Structure during the Restore Operation

In case the include_specification in the Save request had a directory tree structure, then to maintain the same during the Restore operation, you must specify the wildcard "*" in the include specification of the Restore request.

```
/Include = TPRD:[TPRD.APPFILES*]*.*;*
```

If you do not use the wildcard "*" or specify a wildcard other than "*" in the include specification, all the files are automatically restored to the root directory.

```
/Include = TPRD:[TPRD.APPFILES]*.*;*
/Include = TPRD:[TPRD.APPFILES%]*.*;*
```

5.3.3 /NOASSIST Qualifer Cannot be Used with the /PORT Qualifier

The /NOASSIST qualifier cannot be used with the /PORT qualifier when executing the MDMS MOVE VOLUME request using the /PORT qualifier.

5.3.4 /PORT Qualifier Cannot be Used from the MDMS GUI

/PORT qualifier cannot be used with the MDMS MOVE command when this command is issued from the MDMS GUI.

5.3.5 ALLOCATE DRIVE Preceded by LOAD VOLUME Results in Allocation of a Wrong Drive

ALLOCATE DRIVE preceded by the LOAD VOLUME, vol_name/DRIVE=drive_name results in the allocation of a wrong drive. An internal QAR has been raised to resolve this issue and the solution will be provided in the future release. However, this problem is observed when the command is given at the DCL prompt only.

5.3.6 DCL Syntax for Lists must be Followed or You might Get Unexpected Results

If you do not use the correct syntax when specifying lists in MDMS commands, DCL parsing may cause the command to work differently than expected. Becasue MDMS allows multiple objects on all commands, if a list is specified for one of the qualifiers, you must put parentheses ( ) or quotes "" around the lists to avoid any unexpected behavior. For example:

```
$MDMS CREATE DRIVE d1/MEDIA=m1,m2
```

This will create two drives, d1 and m2. No error would be generated. If you use the correct syntax, you will get one drive d1 with two media types. The correct syntax is:

```
$MDMS CREATE DRIVE d1/MEDIA=(m1,m2)
```

Workaround:

Use the syntax as described in the MDMS Reference Guide and in MDMS help.

5.3.7 Motif Look and Feel

An arrow button is provided for the List, Combo, and Date fields in MDMSGUI. The size of the arrow button is same for all the above fields. However, the arrow button size varies in the Motif Look and Feel using Java 1.4 version.

5.3.8 /AFTER_SCHEDULE Logic Misbehaves if ABS is Shutdown when a Save is in Progress

Consider a schedule TEST_SAVE_SCHED corresponding to save TEST. When the schedule is started, MDMS initiates the save, which executes in the context of ABS. When the save completes, ABS updates the status of the save request in the transaction status field. If ABS is shutdown when the save request is executing/initiated, the save status is not updated to the transaction status field (since ABS is not running), which continues to show the status from the previous run.

Becasue the transaction status field continues to have the old status, the /AFTER_SCHEDULE logic uses this old status to schedule the after schedule jobs. Hence, we see the /AFTER_SCHEDULE logic misbehaving.

5.3.9 /AFTER_SCHEDULE Qualifier

There is a restriction with using the /AFTER_SCHEDULE qualifier to link jobs. Only those schedules (created automatically by MDMS) that have an associated save can be assigned to the /AFTER_SCHEDULE qualifier. Schedules that do NOT have an associated save cannot be assigned to the /AFTER_SCHEDULE qualifier.

Hence, any schedule (one with an associated save, or one which executes DCL commands) can have a dependency on a schedule with an associated save, but not on a schedule which executes DCL commands. This is a current MDMS design limitation.

5.3.10 V4.x Stores Object Names in Uppercase Only

ABS V3.x versions allowed both lower and uppercase for object names. ABS /MDMS V4.x supports only uppercase names; that is, the names are stored the names are stored in uppercase in the database. Hence, if you upgrade from V3.2A to V4.x, and you have two drives names "xxx", and "XXX", you may not be able to modify the attributes of the drive named "xxx". Any attempts to change the attributes will affect the drive named "XXX"(uppercase) only.

5.3.11 Fiber Channel Environment

In an FC environment, there is no universal manager to maintain information of devices (such as, which node the device has allocated). Further, in an FC enviroment, VMS does not manage the allocation of drive across two or more clusters. Hence, multiple users can allocate the same drive at the same time from two nodes (they may be either standalone nodes or they may be present in two different clusters). To overcome this, MDMS polls the nodes listed in the drive object to ensure that none of the nodes listed in the drive list has allocated the device. Thus, if MDMS cannot reach a node, it returns the drive check error. MDMS will not allow other nodes to access the drive without knowing the status of the drive on one node as it may possibly lead to a data loss scenario.

This is a design restriction with MDMS and the system administrator must take judicious decision when the node is not reachable for some reasons , such as when a node was shutdown. This restriction may lead to a potential save failure with a drive access error.

A possible workaround to the issue is listed below:

The customer must create an MDMS GROUP object. The GROUP object should consist of all the NODEs accessing the DRIVE/JUKEBOX and the DRIVE/JUKEBOX objects should have the GROUP listed in the DRIVE/JUKEBOX objects. During system bootup, the following command must executed:

```
$ MDMS SET GROUP xyx/NODE=node_name/ADD
```

At the time of the system shutdown the following needs to be executed.

```
$ MDMS SET GROUP xyx/NODE=node_name/REMOVE
```

The above workaround is applicable only when the node is shutting down normally. In case the node is not reachable, when there is a network issue due to reasons other than a normal node shutdown (example: due to a node crash or due to a network cable issue), the above-mentioned workaround will not be applicable.

Another alternative the system administrator can consider is to remove the NODE name from the DRIVE object in case the customer wants to shutdown one of the nodes in an FC environment.

5.3.12 MDMS does not Support Multiple NIC Interfaces with TCP/IP on the same Node

Due to a design limitation, MDMS does not support multiple NIC interfaces on the same node.

6 Documentation Updates and Errata

The following section describes the documentation updates and errata for ABS/MDMS documentation set.

6.1 System Backup to Tape (SBT) for Oracle databases

Oracle 9iR2

The System Backup to Tape (SBT) feature of ABS is qualified to backup Oracle 9iR2 (9.2.0.2) database using Oracle Recovery Manager (RMAN) on OpenVMS Alpha V7.3-2 and higher. Oracle 9iR2 (9.2.0.2) supports shared libraries in RMAN. Hence, unlike prior versions of Oracle there is no need to link the SBT shareable image with Oracle server.

ABS Version 4.5 kit includes the SBT shareable images, MDMS$SBTTEST_MA64.EXE and MDMS$SBTSHR_MA64_9I2.EXE for Oracle9iR2 and later. Sections 9.2 to 9.11 of System Backup to Tape for Oracle Databases, in *HP Archive Backup System for OpenVMS Guide to Operations* are applicable for Oracle 9iR2. Section 9.1, "Linking System Backup to Tape with the Oracle Server" is applicable only for the Oracle releases prior to Oracle 9iR2 and is not applicable for the Oracle 9iR2 and later releases.

Oracle 10gR1

The System Backup to Tape (SBT) feature of ABS is qualified to backup Oracle 10gR1 (10.1.x.x.x) database using Oracle Recovery Manager (RMAN) on OpenVMS Alpha V8.3 and V8.2. All the procedures used in Oracle 9iR2 are applicable to Oracle 10gR1 as well.

Oracle 10gR2

The System Backup to Tape (SBT) feature of ABS is qualified to backup Oracle 10gR2 (10.2.x.x.x) database using Oracle Recovery Manager (RMAN) on OpenVMS Alpha V8.3 and V8.2. and on OpenVMS Integrity servers V8.2-1 and V8.3.

To use SBT feature of ABS, users of Oracle 10gR2 must change the path of SBT_ LIBRARY to SBT_LIBRARY=SYS$SHARE:MDMS$SBTSHR_MA64_9I2.EXE instead of SBT_LIBRARY='abs_sbt', where 'abs_sbt' was a logical defined (in file rdbms_logicals.com) for SBT library in the previous versions of Oracle release.

Following is the RMAN script template:

```
RUN
{
allocate channel t1 type 'sbt_tape'
parms="SBT_LIBRARY=SYS$SHARE:MDMS$SBTSHR_MA64_9I2.EXE,
ENV=(MDMS$SBT_ARCHIVE=ORACLE_DB_ARCHIVE)";
backup <...>
<...>
<...>;
Restore <...>
<...>
 <...>;
release channel t1;
}
```

Except the above changes, all the other procedures used for the Oracle 9iR2 are applicable for Oracle 10gR2 as well.

Note

To perform Duplex backups execute SQL>ALTER SYSTEM SET backup_ tape_io_slaves = true deferred; and restart Oracle Database.

6.2 Quick Reference Guide to Configure ABS/MDMS

The *Quick Reference Guide to Configure ABS/MDMS*, which is intended for all the novice users of HP Archive/Backup System for OpenVMS, is provided along with the ABS kit.

6.3 MDMS BIND VOLUME

In the *Archive Backup System for OpenVMS Operational Guide*, Section 3.3 explains the procedure to update the catalog with the information extracted form the existing saveset from the tape. All procedures, except binding the volume, are explained. The following text explains the procedure for binding the volume:
"If the savesets are present in more than one tape, you need to bind the volumes using the MDMS BIND VOLUME command, so that ABS can receive information about the saveset entries that are present in other volumes."

6.4 Expiration Date and Retention Days

In Section 3.1.7 of the *Archive Backup System for OpenVMS Guide to Operations*, the /RETENTION_DAYS and /EXPIRATION_DATE qualifiers for MDMS CREATE ARCHIVE command are not defined completely. The /EXPIRATION_ DATE and /RETENTION_DAYS qualifier for the archive object are mutually exclusive. As a result, input values can be provided for only one of the qualifiers. Hence, the MDMS SHOW ARCHIVE command is modified to display the qualifier that has been used.
When either of the qualifiers are used, the value for the other qualifier in the MDMS SHOW ARCHIVE command is displayed as "–". If none of these qualifiers are used in the MDMS CREATE ARCHIVE command, a default retention period of 365 days is used.

When /RETENTION_DAYS = 0 is specified while creating an archive using MDMS CREATE ARCHIVE or modifying the attributes of an archive using MDMS SET ARCHIVE, the following message is displayed:

```
%MDMS-I-CONFIRM, SETTING RETENTION TO 0 WILL RESULT IN VOLUMES HAVING
THEIR SCRATCH DATE SET TO THE SAME DAY AS THE COMPLETION OF THE BACKUP.
NO RETENTION OF THE DATA WILL TAKE PLACE. ARE YOU SURE (Y/N):
```

Selecting Y will scratch the data on the same day as the completion of the backup; else selecting N will ignore the command with the following confirmation:

```
%MDMS-I-CMDIGNORED, The previous command was ignored
```

The following table lists the usage of these qualifiers and the appropriate values:

Usage	Input specification	Qualifier specified	As displayed for MDMS SHOW ARCHIVE command	
			EXPIRATION_DATE	RETENTION_DAYS
Definite retention	Date format	/EXPIRATION_ DATE=xx:xx:xx	xx:xx:xx	–
	Number of days	/RETENTION_ DAYS=xxxx	–	xxxx
Infinite retention		/NOEXPIRATION	NONE	–
For ZERO retention		/RETENTION_ DAYS=0	–	0

The following example shows the creation of an archive using /RETENTION=0 and the display of the archive using MDMS SHOW ARCHIVE:

```
$ MDMS CREATE ARCHIVE TEST/RETENTION=0
%MDMS-I-CONFIRM, SETTING RETENTION TO 0 WILL RESULT IN VOLUMES HAVING
THEIR SCRATCH DATE SET TO THE SAME DAY AS THE COMPLETION OF THE BACKUP.
NO RETENTION OF THE DATA WILL TAKE PLACE. ARE YOU SURE (Y/N):y
$ MDMS SHOW ARCHIVE TEST

        Archive: TEST
    Description:
 Access Control: NONE
          Owner: UNO::SYSTEM
   Archive Type: TAPE
        Catalog -
         - Name: ABS_CATALOG
        - Nodes:
  Consolidation -
     - Interval: 0007 00:00:00
     - Savesets: 0
      - Volumes: 0
    Destination:
         Drives:
Expiration Date: --
       Location:
  Maximum Saves: 1
     Media Type:
           Pool:
  Retention Days: 0
    Volume Sets:
```

6.4.1 Errata

- In the *Archive Backup System for OpenVMS Command Reference Guide* (on pages 1-135), the following text for the MDMS MOVE VOLUME /PORT qualifier:

  "All the OPCOMS are displayed once and do not require user acknowledgement."
  stands corrected as
  "This OPCOM displays the status on volume(s) movement and does not require user acknowledgement."

  The MDMS Help is updated appropriately.

- In the *Archive Backup System for OpenVMS Command Reference Guide* (on pages 1-19 and 1-144), the following text for the commands MDMS CREATE ARCHIVE and MDMS SET ARCHIVE:

  "If neither an expiration date nor retention days are specified, then the data is retained indefinitely.
  By default, data is retained for 365 days."
  stands corrected as
  "If neither expiration date nor retention days are specified, then the data is retained for 365 days."

  The MDMS Help is updated appropriately.

- In the *Archive Backup System for OpenVMS Guide to Operations*, the display for Expiration Date in the examples specified for "Creating an Archive" (Section 9.4.2, Page 9-9) and "$ MDMS SHOW ARCHIVE SYSTEM_ BACKUPS" (in Appendix A, Page A-9) should be "–" instead of "NONE".

7 MDMS Configuration and Operation

Once MDMS Version 4.5 is installed and any conversions are performed, you might want to adjust the configuration before performing MDMS operations. A DCL command procedure MDMS$SYSTEM:MDMS$CONFIGURE.COM has been provided for initial installation (not a conversion from V2.9x).

For more information on configuration and operation, see the *Guide to Operations* section of your ABS documentation set.

8 Converting ABS 3.* to Version 4.5

In ABS Version 4.5, the ABS Policy engine is in the MDMS Server. To upgrade to ABS Version 4.5, the ABS 3.x Policy Database information must be exported to the MDMS Database.

Before running the conversion ensure that:

- On the V3.x system, you have done an ABS SHOW object/FULL and kept the listing files. These will be useful when comparing the results of the conversion.

- The old ABS Policy database files (ABS$DATABASE:EPCOT.DB%) are not being currently used.

- ABS Version 4.5 and MDMS Version 4.5 are configured and running on all the Clients and Servers.

Run the SYS$SYSTEM:ABS$CONVERT_V3_TO_V4.EXE utility to convert the ABS 3.x database to the MDMS Database.

The utility exports only the highest version of the policy object. Details of new MDMS objects created are logged in ABS$CONVERT_V3_TO_V4.LOG.

```
Sample Conversion:
    $ RUN SYS$SYSTEM:ABS$CONVERT_V3_TO_V4.EXE
    Enter the Path of ABS V3.x Policy Database Files [ABS$DATABASE:] :
    Converting ARCHIVE Objects...
    Converting ENVIRONMENT Objects...
    Converting SAVE Objects...
    Converting RESTORE Objects...
```

9 How to Remove Version 4.5 ABS/MDMS Files

Rolling back ABS to V3.2 must only be used in EMERGENCY situations. Any save requests done before the upgrade will be lost. Any MDMS database changes done before the upgrade will be lost. If tape volumes have been allocated, deallocated, or re-used before the upgrade, they will not be reflected in the volume database after the rollback. Please use caution when deciding to do a rollback.

Note

Before upgrading to ABS Version 4.5, to be able to get back to the exact V3.2A situation you were at before the upgrade, your catalogs would have to be backed up while in a quiet state. Then, you could restore the V3.2A catalogs and they would not reflect any saves run after the V4.1 upgrade. If you do not do this, there will be entries in the catalog reference tapes may be in a free state in the old database files.

If you have installed ABS Version 4.5 and then need to do a rollback to V3.x:

1. Shutdown ABS and MDMS.

2. Manually deassign the logicals ABS_CATALOG_VAOE_CLEANUP and ABS_V40_LOOKUP in the System table if defined earlier.

3. Execute the following command procedure:

   ```
   $ @ABS$SYSTEM:ABS$REMOVE_V40_FILES.COM
   ```

4. Reinstall ABS/MDMS V3.2n. Ensure that you do not use the STANDARD installation option and that you install both ABS and MDMS. You may optionally restore backups of your ABS catalogs.