Software Product Description

PRODUCT NAME: RMS Journaling for OpenVMS, Version 6.2 **SPD 27.58.10**

Note: This is the Software Product Description (SPD) for the following two products:

- RMS Journaling for OpenVMS VAX
- RMS Journaling for OpenVMS Alpha

Except where specifically noted, the features described in this SPD apply to both products. The license and part number information is architecture specific. Please refer to the Ordering Information section of this SPD for further details.

DESCRIPTION

RMS Journaling for OpenVMS helps maintain the data integrity of OpenVMS RMS files and protects this file data from becoming lost or inconsistent in the event of a number of failure scenarios.

RMS Journaling for OpenVMS provides the following three methods of journaling:

- After-image (AI) journaling provides the ability to redo a series of modifications to a file. This type of journaling helps recover lost or corrupted files. After-image recovery restores the contents of the file from the point of the latest backup copy of that file.
- Before-image (BI) journaling provides the ability to undo a series of modifications to a file. This type of journaling allows a file to be returned to a previous known state. This is useful in the event that a file is updated with erroneous data.
- Recovery unit (RU) journaling helps maintain transaction integrity, where a transaction consists of a group of related operations that must be "atomic". That is, either all of the operations complete in their entirety or none of the operations complete. This type of journaling helps prevent data from becoming inconsistent due to the incomplete execution of a transaction.

Journaling is applied on a file-by-file basis. A file can be marked for after-image, before-image, or recovery unit journaling, or any combination of these methods. Within a given application, any combination of journaling methods can be used.

RMS Journaling for OpenVMS stores the information necessary for data recovery in files known as journals. Multiple files can use the same journal.

RMS Recovery Unit Journaling and DECdtm Transactions

Transactions are defined and managed using the DECdtm transaction services available within OpenVMS. An RMS recovery unit is a set of RMS record operations, performed in the context of a single process, that are part of a transaction coordinated by DECdtm services. RMS recovery units are started automatically by RMS, and RMS recovery units are committed or aborted along with the transaction of which they are a part.

RMS Journaling for OpenVMS functions as a resource manager in transactions that use DECdtm services. DECdtm services are implemented using a two-phase commit protocol as described in the OpenVMS Operating System Software Product Description (SPD 25.01.xx). Recovery unit journaling supports both a one-phase and a two-phase commit protocol.

Remote RMS files (files accessed via the DAP/FAL protocol) that are marked for recovery unit journaling can be modified within a transaction and will be included in the "atomic unit of work" defined by the transaction. Also, the modifications made to RMS files marked for recovery unit journaling can be combined with modifications made using any resource manager that supports DECdtm services into a single "atomic" transaction.

The Recovery Unit Facility (RUF) services have been superseded by DECdtm services. However, the RUF services are still supported and are transparently emulated using the DECdtm services. Existing applications that were written using the RUF services continue to work without recompilation or relinking.

Supported File Organizations

All RMS file organizations are supported by RMS Journaling for OpenVMS. Any sequential, relative, or Prolog 3 indexed file can use journaling. However, the following restrictions apply:

- Prolog 1 and Prolog 2 indexed files are not supported by RMS Journaling for OpenVMS.
- Sequential files have a maximum record size of 32,667 bytes.
- Files marked for recovery unit journaling cannot be modified with the DCL WRITE command.
- Stream files cannot be modified in shared mode in a recovery unit.
- Sequential files having VFC record format do not allow shared access for recovery unit journaling.

When using recovery unit journaling with shared fixed-length sequential files, any abort processing overwrites records added to the recovery unit with zero bytes (null characters). This occurs because the record cannot be deleted from the file.

All files that are marked for journaling must reside on Files–11 Structure Level 2 disks. All journals must also reside on Files–11 Structure Level 2 disks. It is recommended that journals used for after-image journaling reside on a disk volume that is different from the disk volume where the data file that uses the journal resides. Journaling across a network or to a tape device is not supported.

Marking Files for Journaling

RMS Journaling for OpenVMS is enabled on a file-by-file basis. Files marked for after-image, before-image, and recovery unit journaling are enabled by using qualifiers to the DCL command SET FILE. Any combination of qualifiers can be used with a particular SET FILE command or series of SET FILE commands.

The most recent SET FILE command for a given file overrides all previous SET FILE commands for that file.

Successful marking of an RMS file for journaling requires exclusive access to the file specified in the SET FILE command. Exclusive access means the RMS file being marked for journaling must be closed; that is, application programs accessing that file must be shut down before the file can be successfully marked for journaling.

No modifications to application programs are required for long-term (after-image or before-image) journaling. Once a file has been marked for long-term journaling, journal information will be written to the journal each time the file is modified. The file must have been opened after the issuance of the SET FILE command that marked the file for journaling. All modifications to the file are recorded in the journal until the file is unmarked for journaling.

Modifications to application programs are required for recovery unit journaling. At a minimum, the program must specify the start and end of a recovery unit, using DECdtm transactions or recovery unit system services. When a file is marked for recovery unit journaling, all modifications to that file made by the application must be within a recovery unit.

Journal File Maintenance

After-image and before-image journals require periodic maintenance. Because journals can expand indefinitely and a journal must reside on a single volume set, occasional re-marking of a file for journaling and all of the associated operations is required. This implies that applications accessing RMS files marked for after-image or before-image journaling, or both, must be stopped periodically for journal maintenance operations.

Recovering Data — Long-Term Journaling

In the case of long-term (either after-image or before-image) journaling, data recovery is done using the RMS Recovery utility. Recovery is on a file-by-file basis and must be explicitly requested; it is not done automatically. The RMS Recovery utility is invoked at DCL level to either roll forward or roll back changes to the file. The Recovery utility requires exclusive access to the file being recovered. Changes can be rolled forward or rolled backward until a time specified by the user.

Recovering Data — Recovery Unit Journaling

No user intervention is required to roll back a recovery unit that is incomplete. The rollback of an incomplete transaction is started and completed automatically the next time a user attempts to access the file.

Interaction with the Backup Utility

To recover data using after-image journaling, a backup copy of the file must be available. This copy must be made with the Backup utility (BACKUP); a copy of the file made with the COPY or CONVERT command cannot be used. The backup copy of the file must be made after the SET FILE/AI_JOURNAL command is issued but before the file is opened for update. BACKUP requires exclusive access to files being backed up. No user or application program can access the file until the backup is finished.

The use of BACKUP/RECORD is recommended. If the file is then rolled forward, the modifications that have been made since the most recent backup are applied.

If a backup copy of a file is rolled forward with the RMS Recovery utility, users must re-mark the file for after-image journaling with the SET FILE/AI_JOURNAL command. A backup copy of that file must be made after it has been marked for after-image journaling and before application updates are allowed.

A backup copy of the file must be re-marked for journaling if it is to be used in place of the original file.

Interaction with Volume Shadowing for OpenVMS

Volume Shadowing for OpenVMS can be used in conjunction with after-image journaling. After-image journaling helps recover data in the following cases not addressed by Volume Shadowing for OpenVMS:

- Mistaken deletion of a file by a system user or operator
- Corruption of the file system pointers
- RMS file corruption due to a software error or incomplete bucket write operations to an indexed file

Failures Not Addressed by RMS Recovery Unit Journaling

Recovery unit journaling alone does not provide recovery when a multiblock bucket write operation to an indexed file is in progress, leaving the bucket in the indexed file in a corrupt state. Use after-image journaling in conjunction with recovery unit journaling to recover from the following failed multiblock bucket write operations:

- Failure of the VAX or Alpha host during a multiblock write operation, such as a system crash, halt, power failure, or system shutdown.
- Permanent loss of path to the disk during a multiblock write operation.
- Cancellation of a multiblock write operation in progress. This operation is possible only with disks that use the DUDRIVER or that access a disk drive through the MSCP server. Other disk drives ignore the cancellation of I/O and are not affected.

HARDWARE REQUIREMENTS

Refer to the OpenVMS Operating System Software Product Description (SPD 25.01.xx) for hardware requirements and supported processors.

CLUSTER ENVIRONMENT

This layered product is fully supported without restrictions when installed on any valid and licensed OpenVMS Cluster* configuration. The *Hardware Requirements* section of this product's Software Product Description and the OpenVMS Operating System Software Product Description (SPD 25.01.xx) list any special hardware required by this product.

* OpenVMS Cluster configurations are fully described in the OpenVMS Cluster Software Product Description (SPD 29.78.xx) and include CI, Ethernet, DSSI, FDDI, and Mixed Interconnect configurations.

SOFTWARE REQUIREMENTS

RMS Journaling for OpenVMS Version 6.2 is an OpenVMS System Integrated Product that requires OpenVMS Version 6.2.

For additional information, refer to the OpenVMS Operating System Software Product Description (SPD 25.01.xx).

OpenVMS Tailoring

For OpenVMS Version 6.2 the following OpenVMS class is required for full features of this system integrated product:

OpenVMS Required Save Set

For more information about OpenVMS classes and tailoring, refer to the OpenVMS Operating System Software Product Description (SPD 25.01.xx).

GROWTH CONSIDERATIONS

The minimum hardware and software requirements for any future version of this product may be different from the requirements for the current version.

DISTRIBUTION AND INSTALLATION

RMS Journaling for OpenVMS Version 6.2 is a System Integrated Product that is distributed and installed with the OpenVMS operating system Version 6.2.

ORDERING INFORMATION

For RMS Journaling on OpenVMS VAX:

Software Licenses: QL-VDVA*-**
Software Documentation: QA-VDVAA-GZ
Software Product Services: QT-VDVA*-**

For RMS Journaling on OpenVMS Alpha:

Software Licenses: QL-0VHA*-**
Software Documentation: QA-VDVAA-GZ
Software Product Services: QT-0VHA*-**

* Denotes variant fields. For additional information on available licenses, services, and media, refer to the appropriate price book.

SOFTWARE LICENSING

This software is furnished under the licensing provisions of Digital Equipment Corporation's Standard Terms and Conditions. For more information about Digital's licensing terms and policies, contact your local Digital office.

License Management Facility

This system integrated product supports the OpenVMS License Management Facility.

License units for this product are sold on a capacity (per CPU) basis.

For more information about the License Management Facility, refer to the OpenVMS Operating System Software Product Description (SPD 25.01.xx) or the *OpenVMS License Management Utility Manual* in the OpenVMS documentation set.

For more information about Digital's licensing terms and policies, contact your local Digital office.

SOFTWARE PRODUCT SERVICES

A variety of service options are available from Digital. For more information, contact your local Digital office.

SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum of this SPD.

The previous information is valid at the time of this release. Please contact your local Digital office for the most up-to-date information.

©1995 Digital Equipment Corporation.
All rights reserved.

™ The DIGITAL logo, CI, DECdtm, Digital, DSSI, MSCP, OpenVMS, OpenVMS RMS, OpenVMS Volume Shadowing, VAX, VAXcluster, and VMScluster are trademarks of Digital Equipment Corporation.