Software Product Description

PRODUCT NAME: VAX TDMS, Version 1.9 (Terminal Data Management System) **SPD 25.71.15**

DESCRIPTION

VAX TDMS (Terminal Data Management System) is a product designed for the implementation of interactive, forms-intensive applications running on VAX systems. As a terminal subsystem, VAX TDMS can reduce the application development and maintenance effort by replacing application program logic specific to terminal interactions with definitions that are external to the program.

VAX TDMS provides:

- A Screen Editor to define forms that format data on the terminal.
- A nonprocedural language to define the exchange of data between an application program and its associated terminal. These predefined exchanges are called requests and are external to the program.
- Utilities that enable the creation, modification, and storage of form definitions and request definitions.
- A record level programming interface which the application program uses to invoke the predefined requests. The application program calls the program interface, passing it the request name and the program record buffers used in the exchange of data with the terminal.

VAX TDMS applications range from database inquiry/response/update to real-time uses such as the periodic display of an industrial process. VAX TDMS is typically used as a terminal subsystem in terminal data management applications such as order entry, inventory control, distribution and other form-intensive applications.

VAX TDMS not only increases application development and maintenance productivity by providing a separation of terminal management code from application code, but it also promotes end user productivity by providing such features as:

- Vertical Field Traversal - Keys can be defined for rapidly traversing forms vertically rather than having to traverse the form on a field by field basis in the predefined field visitation order.
- Immediate Access to First Field and Last Field - Keys can be defined for immediately moving the cursor to the first field and last field on the form.
- The key function associated with a key can be changed and additional keys can be defined to have other functions. This allows keyboards to be redefined to match the environment of the end user.

VAX TDMS is a fully integrated member of the VAX Information Architecture.

- All VAX TDMS definitions are stored in the VAX CDD/Plus (Common Data Dictionary/Plus). Additionally, VAX TDMS can use record definitions stored in the VAX CDD/Plus by VAX DATATRIEVE, VAX DBMS, or VAX Rdb/VMS. VAX TDMS provides full support for VAX CDD/Plus DMU (Dictionary Management Utility) format definitions and read-only support for CDO (Common Dictionary Operator) definitions.
- VAX TDMS is the standard, integrated terminal manager for VAX ACMS-based applications.
- VAX DATATRIEVE can use VAX TDMS forms to DISPLAY/READ data.
- VAX TDMS supports editing with the VAX Language-Sensitive Editor from within the Request Definition Utility. This editor provides templates that help application developers remember RDU syntax as they develop request definitions.

There are four major steps involved in the implementation of a VAX TDMS application.

digital software

November 1989
AE-P785P-TE

1. Defining the Request

   A request defines what information is displayed at the terminal and what information is collected from the terminal. The VAX TDMS request replaces code that would otherwise have to be designed, written, and debugged in the application program.

   The request identifies the form and the record definitions to be used in the exchange of data with the terminal. INPUT and OUTPUT statements define the mapping between the form fields and the record fields during execution of the request.

2. Defining the Form

   A form definition describes the format of the data that is displayed on the terminal at run time. The form definition also specifies, for each field, what validation procedures are invoked. VAX TDMS field validation includes range checks, list checks and picture validations.

3. Defining the Record

   Data records that are used in a request and VAX TDMS application program must be defined in VAX CDD/Plus. The record definition specifies the type, structure and length of the record that is created by VAX CDD/Plus, VAX DATATRIEVE, VAX DBMS, or VAX Rdb/VMS. During the execution of a request, VAX TDMS performs the necessary data type conversion required to transfer data between the form and the record.

4. The Application Program

   A VAX TDMS application program performs application-specific processing and controls the flow of information between the terminal and the database. The application communicates at a record level with the terminal by calling VAX TDMS to execute requests and it communicates with the database using the subsystem of choice - VAX RMS, VAX DATATRIEVE, VAX DBMS, or VAX Rdb/VMS.

   Applications can be written in any VAX native mode language that adheres to the VAX Procedure Calling and Condition Handling standard. Many languages are able to copy record definitions from the VAX CDD/Plus at compile time. If the application program is written in one of these languages (refer to the VAX Common Data Dictionary/Plus SPD 25.53.xx), then it and the VAX TDMS request can share record definitions in the VAX CDD/Plus. VAX TDMS applications written in languages that do not support such copying must include the record definitions in the application program itself as well as in VAX CDD/Plus.

Components

VAX TDMS is comprised of three major components:

1. The Form Definition Utility

   The VAX TDMS Form Definition Utility (FDU) provides all of the capabilities needed to create or modify form definitions and store them in the VAX CDD/Plus. The VAX TDMS FDU includes a Screen Editor that is used to create a screen image of the form and assign specific attributes. These include form field video attributes, form field validation procedures, and the order in which the input fields should be processed.

2. The Request Definition Utility

   The VAX TDMS Request Definition Utility (RDU) provides all of the capabilities needed to create and modify requests and store them in the VAX CDD/Plus. RDU validates each request to make sure that form and record definitions exist and that all transfers of data between form and record fields are valid. The RDU also builds Request Library Files that the VAX TDMS run-time system accesses during the execution of requests.

   The request library capability is provided to avoid run-time access to the VAX CDD/Plus and thus improve VAX TDMS application performance.

3. Programming Call Interface

   The VAX TDMS programming interface is the means by which an application program executes a VAX TDMS request. A request defines an exchange of data between the program and its associated terminal. The application program calls the program interface and passes it the request name and the program record buffers to be used in the exchange of data with the terminal. This record level interface thus eliminates the need for character level or field level communication with the terminal.

   The Programming Call Interface allows application programs to perform additional operations including writing text to or reading text from the reserved message line on a terminal, enabling or disabling a facility that traces the action of a request, and canceling a request in progress.

   The Programming Call Interface supports both synchronous and asynchronous calls from the application program.

Optional Run-time System

An optional run-time version of VAX TDMS is available. The run-time system allows the execution of applications using VAX TDMS for terminal display and management on a machine other than the one used to develop the application.

Documentation

The VAX TDMS Documentation Set consists of:

- Forms Manual
- Request and Programming Manual
- Reference Manual
- Pocket Guide

HARDWARE REQUIREMENTS

VAX, MicroVAX, or VAXstation configuration as specified in the System Support Addendum (SSA 25.71.15-x).

SOFTWARE REQUIREMENTS *

VMS Operating System

VAX CDD/Plus (Required for development license-only.)

* Refer to the System Support Addendum (SSA 25.71.15-x) for availability and required versions of prerequisite/optional software.

ORDERING INFORMATION

Full Development System

Software Licenses: QL-706A*-**
Software Media: QA-706A*-**
Software Documentation: QA-706AA-GZ
Software Product Services: QT-706A*-**

Run-time System

Software Licenses: QL-711A*-**
Software Media: QA-711A*-**
Software Product Services: QT-711A*-**

* 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 layered product supports the VMS License Management Facility.

License units for both the Full Development System and the Run-time System are allocated on a CPU-capacity basis.

For more information on the License Management Facility, refer to the VMS Operating System Software Product Description (SPD 25.01.xx) or the License Management Facility manual of the VMS Operating System 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.