SPERRY UNIVAC

SERIES 90 - OS/3

PROGRAM PRODUCT SPECIFICATION

Data Base Management System

Type Number: 6218-00

SECTION I

The OS/3 implementation of the CODASYL Data Base Task Group Report of April 1971 and the June 1973 Data Description Language Journal of Development is designated Data Base Management System (DMS). DMS is divided into four major software components:

- Data Description Language – DDL.
- Data Manipulation Language – DML.
- Data Base Management System – DBMS.
- Data Base Utilities.

DMS allows a user to define a data base and various "views" of the data base with the Data Description Language (DDL). Independent users can access the data base concurrently via application programs written in the Data Manipulation Language (DML) and End User Facilities. Access to the data base is controlled by the DBMS which is a reentrant interface between application programs and OS/3. Data Base Utilities are provided to aid in system maintenance. The utilities include modules for loading and dumping of the data base, reporting, printing, initialization, and off-line recovery.

PRODUCT FEATURES

Data Definition Language (DDL)

Logical Data Base Definition

The logical structure of the data base is technically defined as a schema. The schema is created using the COBOL-like description languages, Schema DDL and Subschema DDL. Once the data base has been designed, the Data Administration function commits the design in the language of the DDL. The Schema Processor then processes the source images. Subschemas are produced similarly using the Subschema Processor. It is the set of tables produced by these actions which the DBMS examines to determine the relationships within the data base structure during application program execution.

The DDL is divided into three major sections, the AREA SECTION, RECORD SECTION, and SET SECTION.

Area Section

Area section statements identify one or more areas of the total schema by name and code.

Record Section

As many record types as required can be defined to fit the requirements of the data base design. A different location mode can be specified for each type. The major specifications for any one record are:

- location mode
- calc keys
- whether or not the system is to reject duplicate keys
- the areas and pages in which the records are to be stored
- additional pointers to improve efficiency of the system
- data item descriptions

Item Types

DMS allows the following usage clauses within the DDL:

- DISPLAY. The data item is stored in character format.
- COMPUTATIONAL. The data item is fixed point binary.
- COMPUTATIONAL-1. The data is stored in single-precision, signed floating point mode.
- COMPUTATIONAL-2. The data is stored in double-precision, signed floating point mode.
- COMPUTATIONAL-3. The data item is packed decimal.

The different types of items which can be defined in the host languages are also available to the host language programmer.

Set Section

In the SET SECTION the Data Administrator states the relationship between the record types on an owner/member basis. The description of a particular set includes the following:

- type-automatic or manual, mandatory or optional. If a record is a member of an automatic set, it will be linked automatically by the DBMS when it is stored in the data base. Manual members become part of a set occurrence only when they are the object of an INSERT command. Mandatory record occurrences cannot be removed from a set. Mandatory/optional designations also control the actions of DELETE ONLY and DELETE SELECTIVE statements.
- order
- owner record of the set
- member records of the set (there can be any number of record types defined as member records)
- sort key for sorted sets
- extra pointers to increase set traversal speed

Subschema

The CODASYL DBTG introduced the concept of subschema for COBOL. The subschema provides a correspondence between the schema and the application programs. A subschema defines a subset of the data base. Properly, an application program invokes a subschema which represents only that portion of the data base it is interested in operating on. This reduces the size of the program and perhaps more importantly, makes unavailable those parts of the data base which are irrelevant or restricted.

The object subschema is never directly referenced by the application program. The application program interfaces with the DBMS which decides how to satisfy the application program requests to access the data base.

PHYSICAL DATA BASE DEFINITION

The physical structure of the data base is defined through the data administration function using the Device/Media Control Language (DMCL). The DMCL is divided into four major sections: IDENTIFICATION, DEVICE MEDIA, FILE, and AREA.

The IDENTIFICATION statements identify, the object DMCL module name and the schema upon which this DMCL input is processed.

The DEVICE-MEDIA description statements specify the size of the data base page, the number of data base buffers, the number of space inventory buffers, and the name of the journal file.

FILE descriptions specify the names and sizes of data base files. AREA descriptions establish the relationship between logical areas and data base files.

An AREA can be mapped onto one or more files. Also, some logical data base pages can be reserved for future expansion to an area without being physically allocated to files.

COBOL DATA MANIPULATION LANGUAGE (DML)

The COBOL Data Manipulation Language (DML) provides a COBOL program with a facility for accessing a DMS data base. COBOL serves as the host language of the DML. A programmer may freely intermix DML statements with COBOL source language statements. A preprocessor translates DML statements into a form acceptable to the COBOL processor, which generates the object elements to access a DMS data base. A previously established subschema is also required by the preprocessor, to provide the program with the authorized "view" of the data base.

The procedural commands available in the DML command language are:

IMPART	–	registers the run unit with the DBMS and establishes an interface and loading of the object subschema.
DEPART	–	informs the DBMS of run unit termination and optionally cancels any data base changes made by the run unit (rollback).
OPEN	–	registers with the DBMS the run unit's intent to access one or more areas of the data base and provides the mode of access for each.

CLOSE	–	informs the DBMS of data base areas which the run unit has finished accessing.
FREE	–	releases some or all locks put on the data base by the run unit and optionally establishes a new data base recovery point.
STORE	–	creates a new record occurrence in the data base by acquiring the space and establishing linkages with other records in accordance with the criteria defined for this record type in the schema definition.
FIND	–	establishes currency for a particular record, the selection of which is determined by the record selection clause of this command in conjunction with the schema definition for this record type.
GET	–	retrieves the current record of the run unit and moves it to a user-defined buffer.
FETCH	–	combines the functions of the FIND and GET commands.
MODIFY	–	changes any or all data item values of the current record of the run unit and effects record linkage changes in accordance with the specifications in the schema.
DELETE	–	removes the current record of the run unit from the data base by the surrender of its space in the data base and the removal of all linkages to this record.
KEEP	–	informs the DBMS of the run-unit's intent to access the current record of the run-unit at some later time. The DBMS will then put a lock on the record to prevent other run units from changing this record.
INSERT	–	links the current record of the run unit into sets specified by this command.
REMOVE	–	is the opposite of the INSERT command in that it cancels membership (deletes linkages) of the current record of the run-unit from specified sets.
IF	–	causes a data base condition to be evaluated with subsequent action dependent on whether the value of the condition is true or false. A determination may be made of whether a specified set is empty (no member occurrences) or whether the current record of the run-unit participates as the member of a set.
MOVE	–	passes the contents of a specified currency status indicator to the run-unit as a data base key. The specified currency status indicator may refer to a record that is current of the run-unit, record, area, or set.
BIND	–	establishes an interface of the subschema with the DBMS.
UNBIND	–	relinquishes the interface of the subschema with the DBMS.
ROLLBACK	–	cancels data base changes since the last commit point.

The Data Base Management System supports a collection of alternative storage structures and access techniques. Records may be maintained on the data base in one or more of three storage structures: direct, key transformation and chained set. Records may be processed either randomly or sequentially, based on physical key (data base key), or logical key (symbolic key), to other records.

The DBMS allows concurrent access to a shared data base by multiple users in any combination of batch, time-sharing and transaction programs. Conflict situations between concurrent programs are handled through the use of locks on the data base pages, which typically contain several data records, plus automatic queuing and dequeuing mechanisms, and automatic deadlock detection and program rollback. Data base integrity is provided by logging before and/or after images of altered data base pages to a journal file on tape or disk, and (optionally) before images to a "quick-before-look-file". Quick-before-look images

are used to roll back an executing program's updates in the event of a program error, a system-detected deadlock or other error which precludes a successful termination of the program. In the event of a system failure, quick-before-looks are also used to roll back all executing programs during subsequent restart.

DMS is designed to maintain a high degree of control over interaction with the operating system. This minimizes the risk of errors in Job Control or other system-directing functions, and maintains a high degree of security.

IMS/DMS INTERFACE

The DBMS Data Base Management System is the run time element of DMS. It is reentrant and shareable by both batch and on-line run units. IMS is considered to be a run unit to DMS when transaction processing is performed.

There are three ways in which IMS can utilize the data base:

- First, IMS COBOL action programs can be written with embedded DMS data manipulation language statements to access the DMS data base. This procedure is similar to the method employed by batch programs.
- Second, IMS RPG II, COBOL or ASSEMBLER action programs can be written to access DMS data bases through IMS Defined Record Management. Assuming that the programming logic stays the same, existing transaction programs can be used unchanged. There is no change to user interfaces for the purpose of accessing the DMS change to user interfaces for the purpose of accessing the DMS data base through Defined Record Management.
- Third, operators can manipulate data in the DMS data base through Defined Record Management using UNIQUE. Since the commands are unchanged for the purpose of accessing DMS data bases, operators can use the same commands for accessing either the data base or conventional files.

The data base can be used in its original form or it can be passed through the defined record management, thereby providing all of the item level validity testing and control that is available to users of conventional files. This approach provides a high degree of flexibility in an on-line data base system.

DMS SUPPORT AND SERVICE ROUTINES

The data administration function accomplishes the on-going maintenance of the data base system through the following utility routines:

Page-Initializer Routine:	initializes the DMS data base with empty pages. It can be used to initialize a new data base or a range of pages in an existing data base.
Data Base-Unload Routine:	copies specified pages or areas of the data base to magnetic tape or disk for recovery purposes. A complete analysis of space utilization and the number of occurrences of each record type are also displayed.

Data Base Display and Alter Routine:	displays the contents of a page in hexadecimal and EBCDIC. Data base-key and other control information is included. Also included is a page-patch routine.
Data Base-Reload Routine:	copies a specified range of pages or areas from magnetic tape or disk and writes them into the data base.
On-Line and Off-Line Recovery/Rollback Routines:	restore the integrity of the data base following a software or hardware crash. Also, run units may be rolled back due to user request, DBMS error, or deadlock detection. Off-line recovery is supported to reset the data base backward or forward in time from its current state.
JCL Procedure Library:	included in the DMS system to assist integrating DMS.
Journal File Display:	procedures journal reports.
Journal File Fix Utility:	closes the journal file after a system failure.

SOFTWARE REQUIREMENTS

DMS requires the following OS/3 software products for operation:

- SCS – OS/3 System Control Software
- ESS – OS/3 Extended System Software
- 1974 ANS COBOL, or COBOL-68

HARDWARE REQUIREMENTS

DMS will operate on any 90/25, 30 or 40 model and configuration that meets the minimal hardware configuration requirement for that specific model and satisfies the main storage requirements specified in the software release documentation accompanying each release.

Additional main storage and/or peripheral devices may be required, depending on the user's selection of the system's supported features and the size of the user's programs, files and data bases.

SECTION II

CUSTOMER EDUCATION

Sperry Univac makes available customer education related to this program product. Course availability and schedules are contained in the published course catalog. Charges for courses will be at the then prevailing rates. Customers should contact their local Sperry Univac representatives for enrollment procedures.

PROGRAM PRODUCT SUPPORT

Sperry Univac will endeavor to correct any significant error in an unaltered current release of the Program Product, which the customer brings to the attention of Sperry Univac in accordance with established correction procedures. Sperry Univac does not represent or warrant that all errors will be corrected. This error correction service may result from time to time in update releases which the customer will install. Sperry Univac reserves the right to alter the classification of this Program Product to reflect changes in policy or support requirements.

ORDERING INFORMATION

This Program Product and its associated documentation may be leased from Sperry Univac at separately stated lease charges. Upon execution of a Supplement for Program Products (Form UD1-1306) or its equivalent for this Program Product, the following will be provided:

1. A magnetic tape, diskette(s) or removable disk media in OS/3 Operating System format containing:

   - DMS
   - Installation Verification Program

2. One copy of the associated documentation:

   - Software Release Documentation
   - DMS System Support Functions User Guide and Programmer Reference, UP-8272
   - IMS/DMS Interface User Guide and Programmer Reference, UP-8748
   - DMS Summary, UP-9009
   - DMS Data Manipulation Language User Guide and Programmer Reference, UP-8036
   - DMS Data Description Language User Guide and Programmer Reference, UP-8022