
mv.ENTERPRISE

Online User Reference

Manual, Chapter 1



Release 4.0
August 1997

This edition of the mv. ENTERPRISE manual set is to be used with all mv. ENTERPRISE systems operating under release 4.0 or higher. This electronic document combines the four hard copy volumes into a single file to be viewed with Adobe's Acrobat Reader, 3.01 or later. Content is identical between online and hard copy editions of this manual set.

It is the policy of PICK Systems, Inc. to improve products as new technology, components, software and firmware become available. PICK Systems, therefore, reserves the right to change specifications without prior notice.

Copyright © 1998, 2000 by PICK Systems, Inc.
Irvine, CA 92606
All rights reserved. Printed in U.S.A.

mv. ENTERPRISE is a trademark of PICK Systems, Inc.
MultiValue is a trademark of Spectrum International, Inc.
UNIX is a registered trademark in the USA and other countries, licensed exclusively through X/Open Company, Limited.
SCO and UnixWare are registered trademarks of Santa Cruz Operation, Inc.
AIX is a trademark of International Business Machines Corp.
MP-RAS is a licensed product of NCR Corporation.
PICK is registered trademark of PICK Systems.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Part Numbers: This online document
combines the hard copy
manuals with these part
numbers:

84-00014A01

84-00014A02

84-00014A03

84-00014A04

Software Release: 4.0

Document Release: AA

The material contained in this document is furnished for customer reference only, and is subject to change without notice. PICK Systems, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. PICK Systems assumes no responsibility for any errors that may appear in this document and makes no commitment to update nor to keep current the information contained in this document.

The techniques described here are proprietary and should be treated accordingly. No part of this document may be reproduced in any form or by any means without the prior written consent of PICK Systems.

Note that PICK Systems appreciates receiving suggestions and comments on its publications. Comments may be sent to the publisher:

PICK Systems, Inc.
1691 Browning
Irvine, CA 9606
Tel (800) 367-7425
E-mail: documentation@picksys.com
ATTN: Manager, Technical Publications

This page intentionally left blank.

Chapter 1 Table of Contents

Introduction to mv.ENTERPRISE	1-1
mv.ENTERPRISE Operating System	1-1
System Features	1-3
Virtual Memory	1-3
Architecture	1-4
Input/Output	1-4
mv.ENTERPRISE Assembly Instructions	1-6
mv.ENTERPRISE Instruction Set	1-7
Overview of TCL	1-9
Software Processors	
Software Processors	1-11
File Management Processors	1-11
Editor Processor	1-12
Proc Processor	1-12
ACCESS Processor	1-12
mv.ENTERPRISE BASIC Processor	1-13
Utility Processors	1-14
File Management Overview	1-14
CREATE-FILE Processor	1-14
CLEAR-FILE Processor	1-14
DELETE-FILE Processor	1-15
COPY Processor	1-15
Editor Overview	1-16
Proc Overview	1-17
ACCESS Overview	1-18
mv.ENTERPRISE BASIC Overview	1-20
System Utilities Overview	1-22
mv.ENTERPRISE/Open Data Architecture Overview	1-23
Scribe Overview	1-24

This page intentionally left blank.

Chapter 1: Introduction to mv.ENTERPRISE

mv.ENTERPRISE Operating System

The mv.ENTERPRISE Operating System specializes in data base management. It is a complete system that allows multiple users to instantly update and/or retrieve information stored in the online data files. Users communicate with the system through local or remote terminals to access files that may be private, common, or security-controlled. Each terminal user's vocabulary can be individually tailored to specific application vocabularies.

The system includes the powerful, yet simple-to-use ACCESS inquiry language, the mv.ENTERPRISE BASIC and PROC high-level languages, file maintenance tools, an Editor, complete programming development facilities, and a large variety of other functions. The system runs in an online, multiuser environment; all system resources and data files are efficiently managed by a true virtual memory operating system which provides users with maximum performance and reliability.

The system is exceptional, as measured from any point of view:

- Multiuser performance
- File management languages
- Ease of programming
- Data structure
- Architectural features

Its high performance and fast response times are the result of a unique business-oriented, machine-independent assembly language that greatly reduces system overhead and program execution time.

System software includes the following:

- Virtual memory manager

- Multiuser operating system
- Special data management instructions
- Input/output Processors
- ACCESS, mv.ENTERPRISE BASIC, PROC, TCL languages
- Selectable/automatic report formatting
- Dynamic file/memory management
- Selectable levels of file/data security
- Phantom tasking
- Level pushing

The unique file structure provides the following:

- Variable length files/items/attributes
- Multivalues (and subvalues) within attributes
- Efficient storage
- Fast access to data items
- Selectable degrees of data security
- Unlimited file and item size (up to the size of disk)

System Features

The mv. ENTERPRISE Operating System is specifically designed to provide cost effective data base management application tools. The mv. ENTERPRISE Operating System is a very efficient and effective tool for online data base management. mv. ENTERPRISE is backward compatible with the earlier R83 version. As was true of its predecessors, mv. ENTERPRISE continues as a truly unique online transaction processing system. Three major components of the system are especially important:

- Virtual memory operating system
- Software level architecture
- Terminal input/output routines

Virtual Memory

The first important feature is the virtual memory operating system, long used in larger computer systems, but previously impractical for mini-computers because of the large amount of overhead needed for the operating system itself. In mv. ENTERPRISE, the virtual memory operating system is optimized and coded in a highly efficient, machine-independent assembly language that executes many times faster than conventional languages. Thus, the overhead time is no longer a serious problem on smaller computers.

Most sophisticated computer operating systems require vast amounts of memory to support them. However, only a small amount of main memory is needed to run mv. ENTERPRISE. Everything else (system software, user software, and data) is transferred automatically into main memory from the disk drive by the virtual memory operating system only when required.

Data is organized into frames that are stored on the disk. As a frame is needed for processing, the operating system checks to see if it is already in main memory. If it is not, the frame is automatically transferred from the disk (virtual memory) to main memory. Frames are written back onto the disk on a least- recently-used basis. Virtual memory allows the user access to a programming area not restricted

by main memory, but as large as the entire available disk storage on the system.

Architecture

The second important feature is the software-level architecture, designed specifically for data base management. This architecture includes very powerful instructions for character moves, searches, compares, and all supporting operations to manage variable-length items and subitems. It was designed without the usual restrictions imposed as a result of being tied to any one piece of hardware. This means it is truly a machine-independent approach.

Input/Output

The third major feature is the handling of input/output (I/O) communications with the online terminals. One of the main problems in any online application is that of managing I/O from online, interactive terminals. As these terminals increase in number, the load on the CPU becomes more severe. As a result, the response to the terminals degrades dramatically.

The I/O processing of the online terminals incorporates an overlapped buffering concept. This means that other program execution need not be held up waiting for terminal input/output to complete. As a result, the central processing unit is used more efficiently, resulting in a large number of terminals that may be connected to the mv.ENTERPRISE Operating System before any significant degradation in response time is detected.

System Features

In summary, the system includes these features:

- True data base management
- Comprehensive business computing
- Virtual memory operating system
- Expanded multiuser capabilities
- Online file update/retrieval
- ACCESS retrieval language
- Variable file/item/attribute lengths
- Dynamic file/memory management
- Automatic report formatting
- Total data/system security
- Fast terminal response
- Line-printer spooling
- Special data management processors
- Complete command-stack processor
- High-speed sorting
- Phantom processing
- Level pushing
- Multiple ABS

mv.ENTERPRISE Assembly Instructions

This topic describes the mv.ENTERPRISE Assembly Language's unique structure and operation in terms of architecture and the features of its instruction set. In the early development of the mv.ENTERPRISE Operating System, the task of creating an efficient, flexible business information system was assigned to a team of visionary systems designers. At the time they began, the hardware selection had not yet occurred. While most people might consider this a handicap, it was in fact a fortunate situation. Not being restricted by the limitation of one type of hardware, the designers had the freedom to create a new language, an assembly language optimized for business data processing.

The power and flexibility in this assembly language is the strength of the current mv.ENTERPRISE Operating System. The mv.ENTERPRISE instruction set is specifically designed for character moves, searches, compares, and all supporting operations related to managing variable length items and attributes.

The virtual machine has 16 addressing registers and one extended accumulator for each terminal. A return stack accommodates up to eleven subroutine calls for each terminal. By indirect addressing through any one of the 16 registers, any byte in the virtual memory can be accessed. Relative addressing is also possible using an offset displacement plus one of the 16 registers to any bit, byte, word (16 bits), double-word (32 bits), triple-word (48 bits) or quadruple-word (64 bits) in the entire virtual memory. This ensures fast response time and very high system throughput.

mv.ENTERPRISE Instruction Set

Features of the mv.ENTERPRISE Operating System instruction set include:

- Bit, byte, word, double-word, triple-word, and quadruple-word operations
- Memory-to-memory operation using relative addressing on bytes, words, double-words, and triple-words
- Bit operations permitting the setting, resetting, and branching on condition of a specific bit
- Branch instructions permitting the comparison of two relative memory operands and branching as a result of the comparison
- Addressing register operations for incrementing, decrementing, saving, and restoring addressing registers
- Byte string operations for moving of arbitrarily long byte strings from one place to another
- Byte string-search instructions
- Buffered terminal input/output instructions
- Handling of all data and program address references by the virtual memory operating system
- Conversion of binary numbers to printable ASCII characters and vice versa
- Arithmetic instructions for loading, storing, adding, subtracting, multiplying, and dividing the extended accumulator and a memory operand
- Control instructions for branching, subroutine calls, and program linkage
- Efficient stack operations for use by high-level languages

For further details regarding the mv.ENTERPRISE instruction set, refer to the *Pick Assembly Language Manual*.

Overview of TCL

Terminal Control Language (TCL) is the primary interface between the terminal user and various system processors. Users invoke most processors directly through Terminal Control Language (TCL) by single input statements and return to TCL after processing completes. TCL prompts the user by displaying a colon (:). Alternatively, by invoking a command-stack processor, a user can collect, save, and reuse previously typed commands. The command-stack processor displays the +: prompt. Both prompts are referred to as the TCL prompt character.

Input statements are entered by typing one character at a time from the terminal until ENTER is pressed, at which time TCL processes the entire line. Each TCL input statement must begin with a valid verb, and can be up to 300 characters in length.

One of the many powerful features of the system is the ability to customize the vocabulary for each user. Since verbs reside in the individual user's master dictionary (MD), the vocabulary may be added to or deleted from the MD without affecting other users. Additionally, create an unlimited number of synonyms for each verb.

The system operates in an echo-plex environment. This means each data character input to the terminal is sent to the computer and echoed back to the terminal before displaying on the screen. The user is thus assured that if the data character displayed on the terminal is correct, the data character stored in the computer is also correct.

Besides recognizing the standard ASCII (96) character set, the system performs special operations when certain additional control characters are detected. All other control characters are deleted from the input line that is passed to lower level processors.

Section 1: Software Processors

Software Processors

The processors available on the mv. ENTERPRISE Operating System comprise the most extensive data base management software available on any computer. An overview of some of the processors available to all terminal users is presented in this topic. Throughout this manual individual topics are devoted to detailed discussions of each of these processors. The following processors, and any other software processors not mentioned here, may be used by any or all terminals simultaneously. Processing is invoked through appropriate verbs contained in each terminal user's master dictionary. However, user accessibility to the full range of processor functions may be limited by controlling the verbs available to specific users.

File Management Processors

The file management processors generate, manage, and manipulate portions of, or entire files within the system. These processors include:

- CREATE-FILE
- CLEAR-FILE
- DELETE-FILE
- COPY
- CREATE-ACCOUNT
- DELETE-ACCOUNT
- PUT and TAKE
- TAKEX and PUTX

Editor Processor

The Editor permits online interactive modification of any item in the data base. The Editor may be used to create and/or modify mv.ENTERPRISE BASIC programs, Procs, assembly source, data files, and file dictionaries. The Editor uses the current line concept; that is, at any given time there is a current line that can be listed, altered, deleted, etc. Editor features include:

- Absolute and relative current line positioning
- Merging of lines from the terminal items or from other file items
- Character string locate and replace
- Input/output formatting

PROC Processor

The PROC processor allows the user to prestore a complex sequence of operations which can then be invoked by a single- word command. Any sequence of operations that can be executed from the terminal can also be prestored by using the PROC processor. The PROC processor includes the following features:

- Argument passing
- Interactive terminal prompting
- Conditional and unconditional branching
- Pattern matching
- Free-field and fixed-field character moving

ACCESS Processor

ACCESS is an information management and data retrieval language. A typical ACCESS inquiry consists of a relatively freeform sentence containing appropriate verbs, file names, data selection-criteria, and control modifiers. ACCESS is a dictionary-driven language. The vocabulary used in composing an ACCESS input sentence is contained in several dictionaries. Each user's vocabulary can be

Software Processors

individually tailored to a particular application terminology. ACCESS features include:

- Logical English word order and syntax for user inputs
- Automatic or user-specified output formatting
- Sorting and statistics generation
- Relational and logical operations
- Verbs such as: LIST, SORT, SELECT, COUNT, STAT, etc.

mv.ENTERPRISE BASIC Processor

mv.ENTERPRISE BASIC is an exceptionally powerful, yet simple and versatile programming language suitable for describing a wide range of processing tasks. mv.ENTERPRISE BASIC is an especially easy language for the beginning programmer to master. As an extended version of Dartmouth BASIC, mv.ENTERPRISE BASIC includes the following features:

- Flexibility in selecting meaningful variable names
- Complex and multiline statements
- String handling with variable-length strings
- Integration with data base file access and update
- Fully structured programming support
- Re-entrant and recursive abilities
- TCL command execution from mv.ENTERPRISE BASIC

Utility Processors

Many utility processors are also included that provide system status and maintenance control.

File Management Overview

The data base management processors generate, manage, and manipulate files (or portions of files). These and other file management processors are described further in Chapter 2: “File

Management”. CREATE-FILE, CLEAR-FILE, DELETE-FILE, and COPY are described briefly below.

CREATE-FILE Processor

The CREATE-FILE processor generates new dictionaries and/or data files. It creates file dictionary entries in the user's master dictionary (MD) and reserves disk space for the dictionary and data portion of the new file. The user needs only to specify the name of the file and a value for the modulo.

The modulo parameter is selected to balance storage efficiency with accessing speed, based on the number of items in the file, the average item size, etc. Required file space is allocated from the available file space pool. Upon demand, files automatically grow beyond their initial size because the system automatically attaches additional *overflow* space from the available file space pool.

CLEAR-FILE Processor

The CLEAR-FILE processor clears the data from a file. At the same time, overflow space that is linked to the primary file space is released to the available file space pool. Either the data section or the dictionary section of a file may be cleared.

DELETE-FILE Processor

The DELETE-FILE processor deletes a file. All allocated file space is returned to the available file space pool. Either the data section or the dictionary section (or both) of the file can be deleted.

COPY Processor

The COPY processor copies an entire file (or selected items from a file) to the terminal, line-printer, another file (in the same account or in some other user-account), or to the same file under a different name (item-ID).

Editor Overview

The Editor is a processor that permits online interactive modification of any item in the data base. The Editor is described further in the “Editor” chapter. The Editor may be used to create and/or modify mv.ENTERPRISE BASIC programs, Procs, paragraphs, phrases, sentences, assembly source, data files, and file dictionaries. The Editor is invoked with the EDIT verb.

Editor commands are one and two-character mnemonics. Command parameters follow the command mnemonic.

The Editor uses the current line concept, that is, at any given time there is a current line (i.e., attribute) that can be listed, altered, deleted, etc.

The Editor includes the following features:

- Two variable-length temporary buffers
- Absolute and relative current-line positioning
- Line number prompting on input
- Merging of lines from the same or other items
- Character string locate and replace
- Conditional and unconditional line deletion
- Input/output formatting
- Prestoring of commands
- Reaffirming the deletion for changed items

PROC Overview

Another system feature is the ability to define stored procedures (Procs). The PROC language is described further in the “Procs and Paragraphs” chapter. The PROC processor allows the user to prestore a complex sequence of terminal control language (TCL) and associated processor operations, which can then be invoked by a single command. Any sequence of operations that can be executed by TCL can also be prestored by using the PROC processor. This prestored sequence of operations (called a Proc) is executed interpretively by the PROC processor and therefore requires no compilation phase.

The PROC processor includes the following features:

- Four variable-length I/O buffers
- Argument passing
- Interactive terminal prompting
- Extended I/O and buffer control commands
- Conditional and unconditional branching
- Relational character testing
- Pattern matching
- Free-field and fixed-field character moving
- Optional command labels
- User-defined subroutine linkage
- Inter-Proc linkage

ACCESS Overview

ACCESS is a user-oriented, data retrieval and information management language for accessing files within the system. ACCESS is described further in Chapter 8, "ACCESS". A typical ACCESS inquiry consists of a relatively freeform sentence containing appropriate verbs, file names, data selection-criteria, and control modifiers. Each user's vocabulary can be individually tailored to particular application jargon.

ACCESS is a dictionary-driven language. The vocabulary used in composing an ACCESS sentence is contained in several dictionaries. Verbs and file names are located in each user's master dictionary (MD). User-files consist of a data section and a dictionary section. The dictionary section contains a structural definition of the data section. ACCESS references the dictionary section for data attribute descriptions. These descriptions specify attribute fields, functional calculations, inter-file retrieval operations, display format, and more.

ACCESS provides the ability to selectively or conditionally retrieve information and to automatically generate reports. Output reports may be displayed to the terminal screen or sent to the line printer. The system automatically formats the reports according to the user-supplied output specifications. In addition, the output may be sorted into any sequence defined by the user.

In summary, ACCESS provides the following functions:

- Relatively freeform input word order and syntax
- Various automatic and user-specified output formats
- Data selection by logical and arithmetic relationships
- Sorting on variable number of sort-keys
- Complex processing of specially created lists
- Manipulation of variables obtained from data files
- Manipulation of subitems within items
- Generation of statistical information regarding files
- Support of signed arithmetic

mv.ENTERPRISE BASIC Overview

The mv.ENTERPRISE BASIC language is an extended version of Dartmouth BASIC, specifically designed for data base management. It is an extremely powerful, yet versatile programming language suitable for addressing a wide range of programming challenges, described in more detail in the “mv.ENTERPRISE BASIC” chapter. Additional features include:

- Optional alphanumeric statement labels of any length
- Alphanumeric variable names of any length
- Multiple statements on one line
- Complex and multiline IF statements
- Formatting and cursor control
- String handling with variable length strings
- Magnetic tape input and output
- Fixed-point, floating-point and string arithmetic
- ACCESS data conversion
- mv.ENTERPRISE file access and update to subvalue level
- Pattern matching
- Dynamic file arrays
- External subroutines
- TCL command execution
- File-level and group locks
- Computed GOTO statements
- Priority CASE statement selection
- Direct and indirect calls
- Alternate compiler emulation

System Utilities Overview

Utility processors perform a wide variety of system tasks, and are described throughout this manual. The mv.ENTERPRISE Operating System includes the following utilities:

- Magnetic tape unit functions
- Mathematical functions
- Line-printer spooling control
- File save/restore functions
- File statistics
- Creating user-accounts
- Setting terminal characteristics
- Block printing
- Virtual memory dumping
- Interuser message communications
- Bootstrapping and coldstarting
- Systems accounting
- Level pushing
- Phantom processing
- Command-stack processing

mv.ENTERPRISE/Open Data Architecture Overview

mv.ENTERPRISE's Open Data Architecture (ODA) provides a standard mechanism for handling remote data sources as if they were local MultiValue files.

The ODA high-level interface through *Remote Q-Pointers* indicates the host name and some identifier of the data in the remote environment.

The Open Data Architecture uses specified host information to access one of several low-level drivers which opens a direct channel into the remote environment. All standard mv.ENTERPRISE file system calls are translated into a limited number of internal commands which all low-level drivers understand and can translate into the appropriate operation in their respective environments.

The ODA interface includes two drivers:

1. mv.ENTERPRISE remote file driver which provides access to remote MultiValue files. The remote mv.ENTERPRISE machine may be a different mv.ENTERPRISE virtual on the same hardware, or a mv.ENTERPRISE virtual on a remote system accessible over a TCP/IP network.
2. Unix file driver which provides access to Unix files. Special files (devices, pipes, etc.) can also be specified with some restrictions.

✓NOTE ODA is licensed as a separate product. Additional information regarding this product is available by contacting PICK Systems Sales.

Scribe Overview

The Scribe Guide is a tutorial and reference manual that gets you started and keeps you going with the Scribe Word Processor.

This guide teaches the basic skills to begin using Scribe and discusses the advanced techniques that make Scribe a full-featured word processor. This dual approach allows you to gain skill in using the fundamentals of Scribe before you learn new skills on an as-needed basis.

The Scribe chapter also includes its own set of appendices which include Problem Solving, Key Functions Chart, Key Functions List and a Scribe-specific glossary and index.

This page intentionally left blank.

