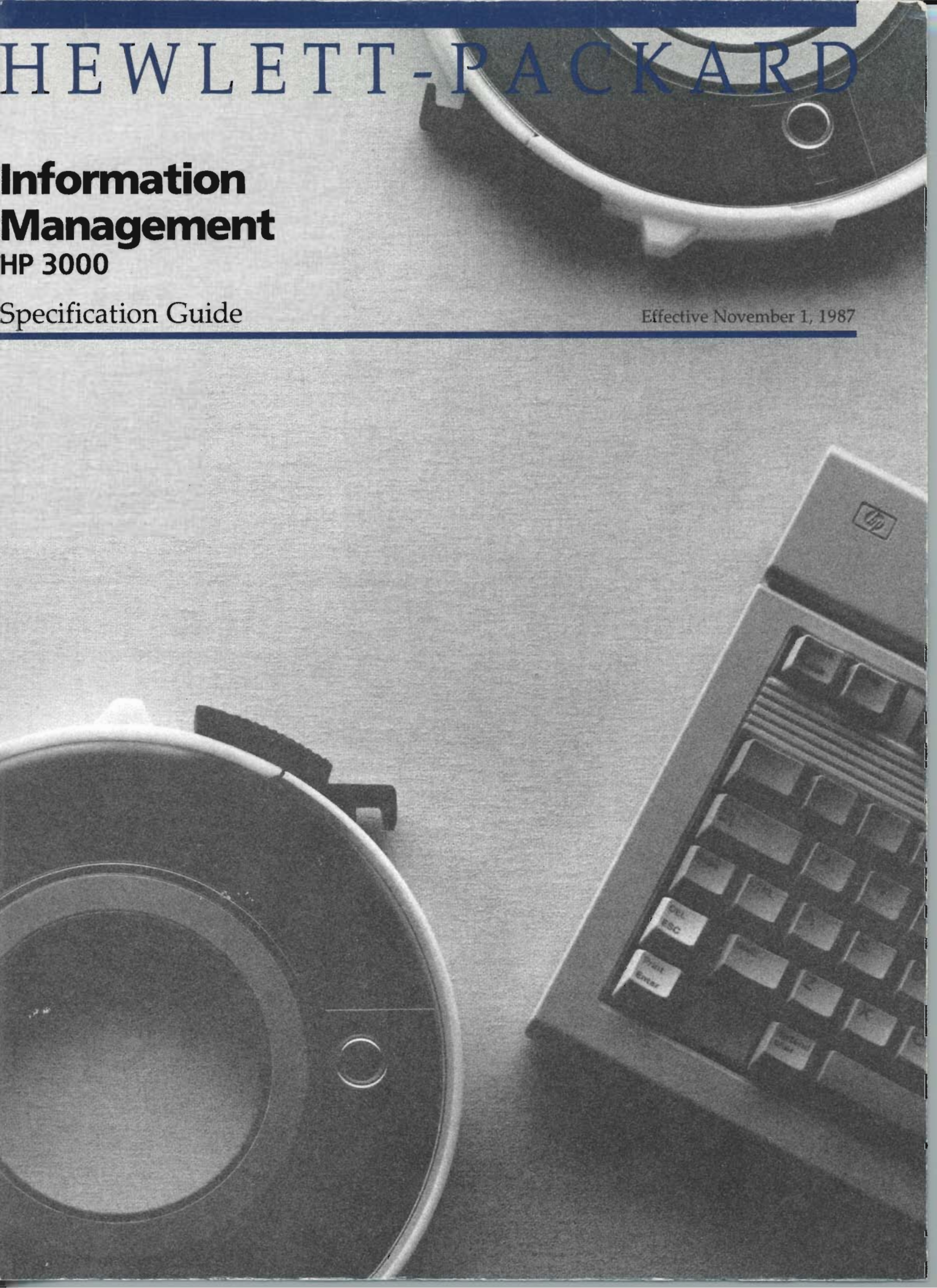


HEWLETT-PACKARD



Information Management HP 3000

Specification Guide

Effective November 1, 1987

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Preface



No two companies are identical. We operate in different industries and we all pursue our own strategies for success. Yet, for all our individuality, we share a common denominator: The challenge of managing information in the most effective way possible. Business systems vary from organization to organization, but each of them is implemented using the same four basic information management components.

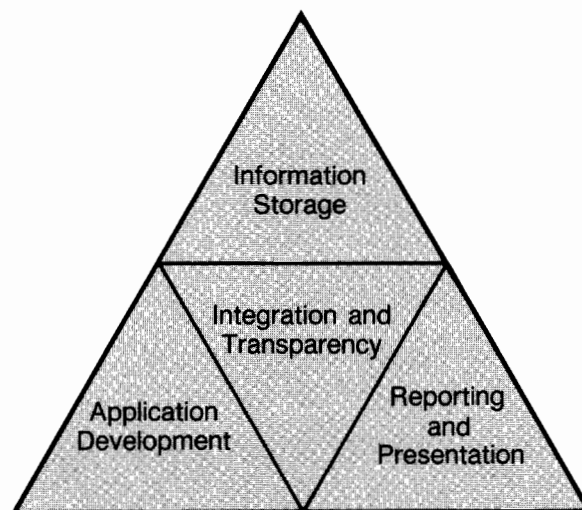
Information Storage. A sound foundation is required to ensure that data integrity is scrupulously maintained while providing flexible data access by applications and users. The optimal foundation will be a relational model DBMS for some applications, a network model DBMS for others, and keyed sequential files for still others.

Reporting and Presentation. A successful business system requires that managers have access to information on their terms. Presented in the format they want. Available when they want. This need will be met in some cases by a programmer using a powerful report writer to quickly create frequently used reports. In other cases, the need will be met by a user friendly query facility which allows the manager or his assistant to directly access the information required to support a split-second decision.

Application Development. The programmer needs a rich environment of tools to productively generate the applications which form the backbone of any business system. A fourth generation language is required to quickly bring the bulk of applications into production. Third generation languages such as COBOL are required for specialized routines where run-time performance is most critical.

Integration and Transparency. As every successful MIS manager knows, a key ingredient to effective business systems is the behind the scenes integration which frees the end user and the programmer from the need to specify endless details when developing applications or requesting information. A sophisticated data dictionary plays a critical role in unifying all of the other components of the business system.

Information Storage. Reporting and Presentation. Application Development. Integration and Transparency. Each area is critical in its own right. Combined they form the big picture, the information management environment that runs the company. The Information Management Framework, shown below, will be used as a vehicle to illustrate how the rich set of information management products from Hewlett-Packard delivers no-nonsense solutions to these needs.



A Note on Product Name Suffixes

You will notice that the names of the products described in this booklet end with either a "V" or an "XL" suffix (for example, COBOL II/V and COBOL II/XL). These suffixes have been added to simplify your selection of the appropriate offerings for your system. Products which have a "V" suffix are designed for use of MPE V based HP 3000 systems. The System Environment section of the data sheet for each "V" product indicates whether it may also be used in Compatibility Mode on the 900 Series of HP 3000 systems. Products which have an "XL" suffix have been specifically designed for use on the 900 Series of the HP 3000 systems.

Table of Contents

Information Storage

HP 3000 Databases at a Glance	1
ALLBASE/XL (HP SQL and HP IMAGE)	3
HP SQL/V	11
TurboIMAGE/XL	17
TurboIMAGE/V	23
TurboIMAGE DBchange/V	29
TurboIMAGE Profiler/V	31
KSAM/V	33
Silhouette/V	35

Reporting and Presentation

Business Report Writer/V	39
Report/V	43
Inform/V	45
QUERY/V	49
HP Visor	53

Application Development

VIRTUOSO Code Generator/V	59
Transact/V	63
Transact/XL	67
Toolset/V	69
Toolset/XL	71
HP's Screen Management Facility	73
Compilers	
COBOL II/V	77
COBOL II/XL	81
HP FORTRAN 77/V	85
HP FORTRAN 77/XL	87
FORTRAN 66/V	89
Pascal/V	91
HP Pascal/XL	93
HP Business BASIC/V Interpreter and Compiler	95
BASIC/V	99
SPL/V	101
RPG/V	103
HP C/XL	105
HP Symbolic Debugger/XL	107

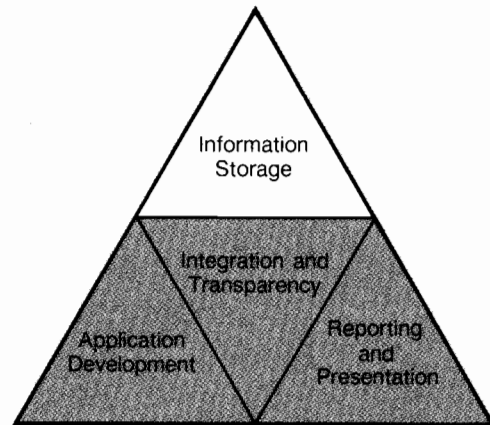
Integration and Transparency Services

HP Dictionary/V	111
HP System Dictionary	115
Information Access	121

System Management Tools

TurboSTORE/V	127
HP Security Monitor/V	129

Information Storage

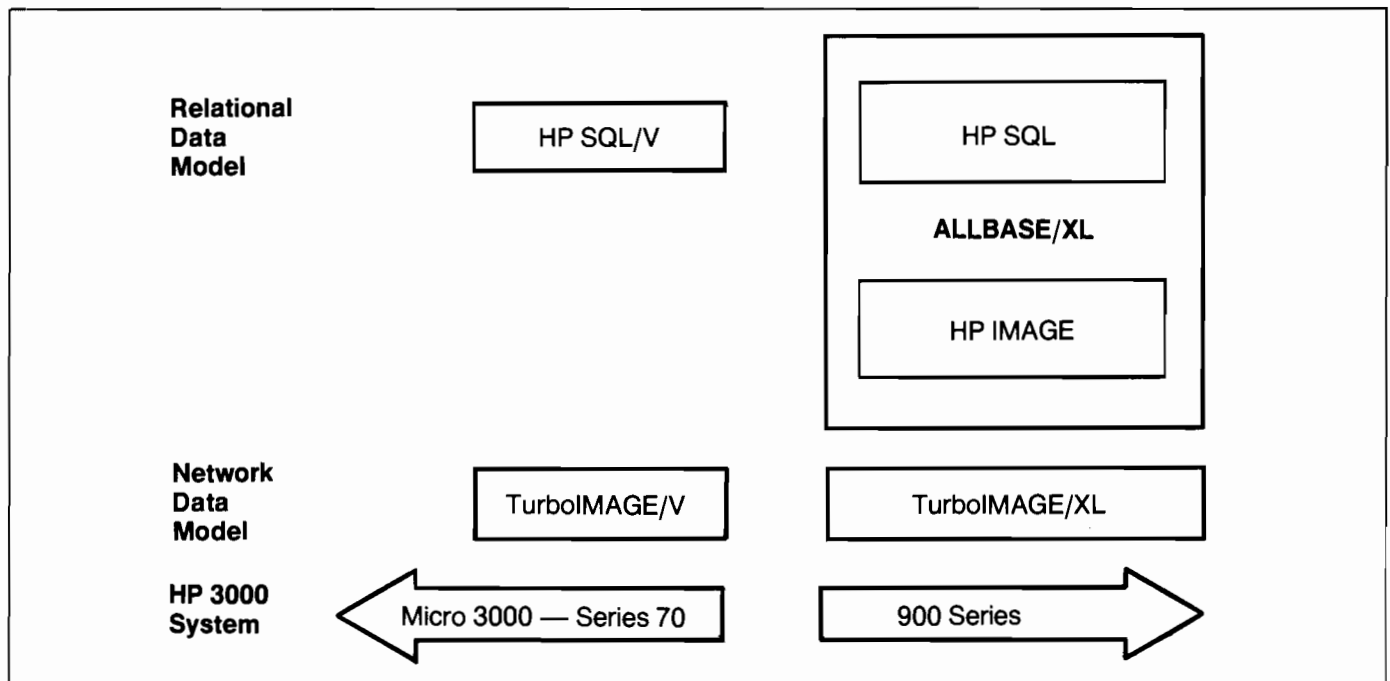




HP 3000 Databases at a Glance

Hewlett-Packard offers compatible relational and network model databases across the entire HP 3000 family. TurboIMAGE/V is the network database product for MPE/V based systems. A fully compatible network database, TurboIMAGE/XL, is available on MPE XL based systems. This provides for application portability across the entire range of HP 3000 computers. HP SQL/V is the relational model database product available on MPE V based systems. For the 900 Series, the ALLBASE/XL product provides an integrated solution consisting of both a relational model interface (HP SQL), and a network model interface

(HP IMAGE). The HP SQL interface of ALLBASE/XL offers compatibility with the HP SQL/V product, insuring applications portability across systems. The first release of ALLBASE/XL will contain the HP SQL interface. Future releases will add the HP IMAGE interface, and the capability to access HP IMAGE data structures via HP SQL. This dual access capability will allow application developers to take advantage of the benefits of both the relational and network database models. The data sheets which follow describe each of these products in greater detail.



HEWLETT-PACKARD

Product Number 36216A

ALLBASE/XL

For the 900 Series of HP 3000 Systems

Making It Easy to Have It All

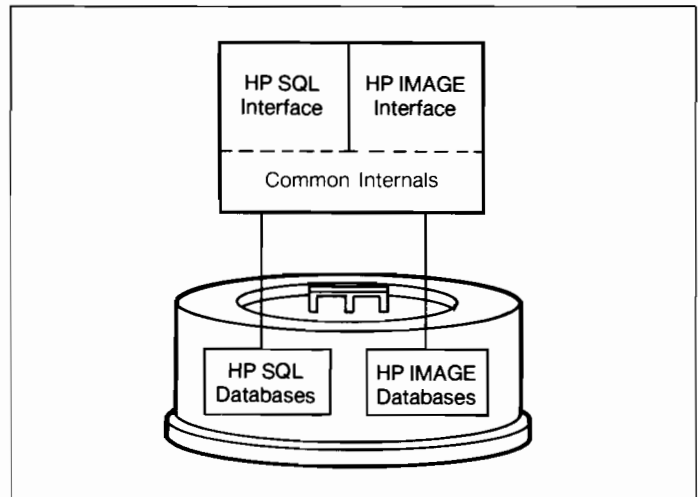
ALLBASE is the leading-edge database management system (DBMS) which lets you choose the appropriate data model on an application by application basis. With ALLBASE's integrated relational and network data models, you are not forced to fit all of your applications to a single data model, relational or network. Since you are now provided with a choice, the decision of which model to use depends solely on which model provides the optimal foundation for each application.

ALLBASE offers a comprehensive set of functionality whether using HP SQL, the relational model interface, or HP IMAGE, the network model interface. There are many additional features tailored to the specific needs of each interface which may be used with the databases created using that interface. A future release of ALLBASE will allow HP IMAGE databases to be accessed via HP SQL.

ALLBASE is built on a solid foundation of common internals which have been designed specifically to exploit the Native Mode performance of the 900 Series of HP 3000s.

Key ALLBASE Features

- Coexisting relational model interface and network model interface allow the optimal data model to be chosen on an application by application basis.
- Data may be accessed directly from a terminal or via application programs written in common programming languages.
- Security is maintained by allowing specification of appropriate levels of access privileges to individual users or groups of users.
- Concurrent access allows multiple users to access data simultaneously.
- Data independence allows changes to be made to the database structure without requiring that applications be modified.
- User-controlled transactions ensure that data is always in a consistent state.
- Automatic locking ensures data integrity in a multi-user environment by preventing access to data while it is being updated.
- Automatic (rollback) recovery guarantees logical data integrity in the event of a soft crash.
- The rollback capability may also be invoked in a program to allow erroneous data, usually generated in an on-line situation, to be purged before the transaction is completed.



- Rollforward recovery guarantees logical and physical data integrity in the event of a hard crash.
- Fast data access is supported through B-tree indexes (HP SQL and HP IMAGE) and hashing (HP IMAGE only).

HP SQL Specific Features

- Relational data model allows the user to specify only what data is required, without specifying how to retrieve it.
- SQL relational language for data definition and manipulation is the industry standard.
- Views may be defined which allow a group of users to view parts of one or more tables as a single, virtual table customized to their particular needs.
- Interactive SQL (ISQL) allows the database to be queried from a terminal, using the powerful SQL relational language.
- Language preprocessors allow the same statements which are used for a direct query via a terminal to be embedded in COBOL or Pascal application programs.
- A sophisticated query optimizer relieves the programmer of the details of query planning.
- Dynamic restructuring allows the data structure, table capacities, and security to be changed without unloading and loading the database.
- Null data values allow use of fields which are relevant to some, but not all, records in a table.
- Application programs are source code compatible with HP SQL/V.

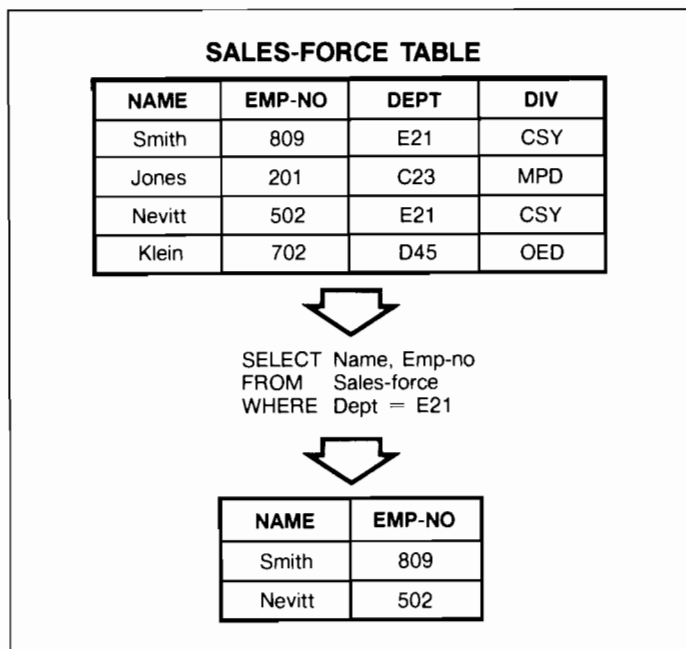
HP IMAGE Specific Features

- Network data structure allows fast access to complex relationships among data.
- Multiple-level relationships allow hierarchical data structures to be easily modeled.
- IQUERY provides the programmer and database administrator with a powerful query facility.
- High-level intrinsics allow applications written in COBOL, FORTRAN or Pascal to easily access the database.
- Dynamic capacity expansion and security modification allow data set capacities to be expanded and security designations to be modified without unloading and loading the database.
- Generic search allows records to be located by specifying only a portion of the value in a field.
- Referential integrity may be maintained through definition of parent/child relationships.

Functional Description

HP SQL: A Complete Relational Model Interface

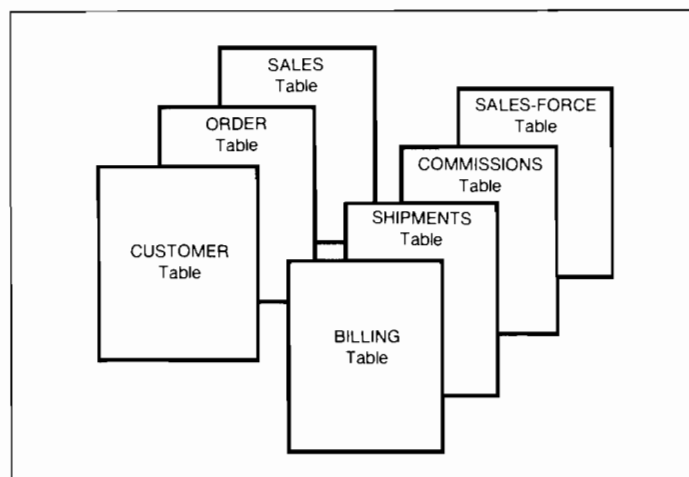
HP SQL is a functionally complete, relational model interface based upon the industry standard SQL relational language. SQL is a nonprocedural language whose powerful commands operate on entire sets of data at a time, rather than the one record-at-a-time procedural approach of network and hierarchical model databases.



The Set-at-a-Time Query Approach of HP SQL

The select, project, and join operators of relational theory are supported. These are implemented through the SQL language with the SELECT command for reading data. HP SQL also supports the other three data manipulation language (DML) commands of the SQL language: INSERT, for inserting groups of records; UPDATE, for updating (modifying) groups of records; and DELETE, for deleting groups of records.

HP SQL does not require that explicit relationships between different tables (data sets) be defined. Relationships are determined at the time a query is performed by matching values between common columns in two or more tables. Since data from any number of tables which share a common column may be related as needed, an HP SQL database consists of a group of tables which the user decides he may have a need to relate.



Simplified HP SQL Relational Structure

A sophisticated query optimizer determines the optimal data access strategy based on such factors as the presence of indexes and the relative sizes of the tables being accessed. This frees the programmer from detailed query planning without sacrificing performance.

Language preprocessors provide easy, high-level access to the database via common programming languages. This approach significantly improves performance by allowing query optimization to be performed when the application is compiled instead of at run time.

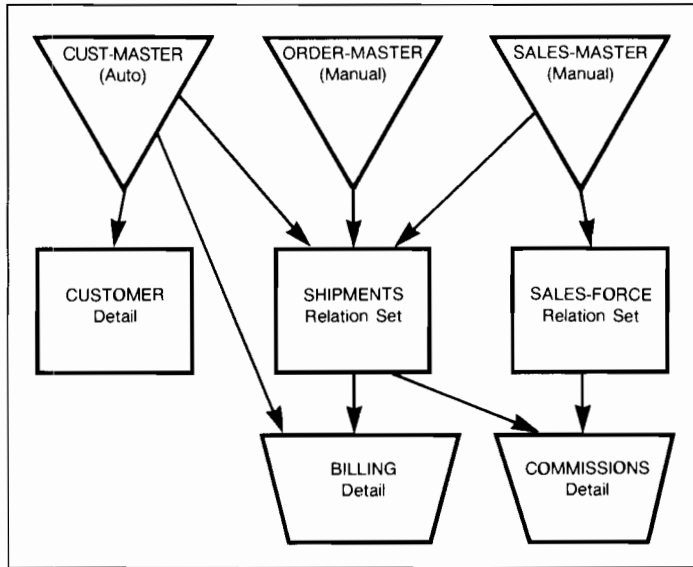
Interactive SQL (ISQL) is the comprehensive interactive interface to HP SQL. There are two distinct sets of functionality which may be used via ISQL. The first is the data definition language (DDL) which allows the database administrator (DBA) to control all aspects of database creation and modification.

The second set of ISQL accessible functionality is a complete implementation of the SQL data manipulation language which allows the programmer or frequent user to query the database from a terminal using the nonprocedural SELECT, UPDATE, INSERT, and DELETE commands.

HP IMAGE: A Complete Network Model Interface

HP IMAGE is a functionally complete, network model interface which continues the tradition of the DATAPRO award-winning IMAGE while incorporating new functionality. With HP IMAGE, relationships between data sets are predefined as part of the database structure (called the schema). These predetermined relationships provide application programs with the fastest data access ALLBASE has to offer. HP IMAGE allows multiple levels of parent/child relationships to be defined. A parent/child relationship, for example, exists between an order data set (parent) and a shipments data set (child) since a valid shipment can only be made against an existing order.

A rich set of relationships may be defined using HP IMAGE: a parent set may have multiple children; a child set may have multiple parents; and a set may be both a parent to some sets and a child to others (multiple levels). With these relationships, HP IMAGE can directly model the information flow of your business.



Simplified HP IMAGE Network Structure

HP IMAGE implements its data manipulation language with a full set of reading, inserting, updating, and deleting capabilities which may be exercised from an application program via intrinsics (library routines).

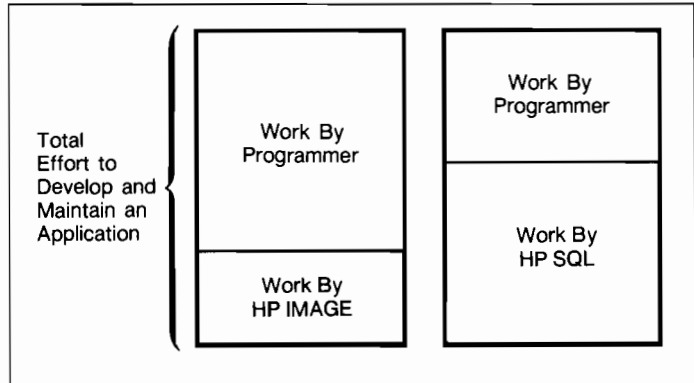
A comprehensive set of data definition language commands and utilities for defining and maintaining the database are provided by HP IMAGE.

HP IMAGE also includes IQUERY, which is a powerful query facility for the programmer and a database administrator (DBA).

**When to Use HP SQL —
When to Use HP IMAGE**

ALLBASE includes a relational model interface and a network model interface because both data models have their place in meeting your business needs.

Applications may be developed in less time with HP SQL than with HP IMAGE. With HP SQL, the programmer must specify only what data is required, not how to access it. This significantly reduces the amount of query preplanning required during application development and maintenance. The nonprocedural nature of SQL also results in greater data independence and less application code. The high degree of data independence allows the database to be restructured without requiring that applications be modified. The reduction in code also improves the productivity of maintenance programmers by allowing the application code to be quickly understood and modified when changing business needs require that applications be updated.



Programmer Productivity Advantage of HP SQL

The procedural HP IMAGE interface allows the programmer to specify exactly how the data is to be accessed. While this requires more query preplanning, it allows the sophisticated programmer the control necessary to obtain optimal performance to support throughput or response-time critical applications.

A future release of ALLBASE will allow HP IMAGE databases to be accessed via HP SQL, although not vice versa. Thus, databases which you will eventually want to access via both interfaces should be created using HP IMAGE.

ALLBASE: Ensuring Data Integrity

Views

HP SQL allows views to be created. A view is a table derived by defining a "filter" over one or more tables to let users or programs view only certain data in the tables. Views improve security by allowing users to access only the data they have a need to know. Since the view is not actually a physical table, use of views does not result in redundant data. When data in a table is updated, all views which use that data are automatically updated.

Security

Security is maintained by allowing specification of appropriate levels of access privileges to individual users or groups of users.

HP SQL allows read (SELECT) access and write access privileges to be assigned at the table level. Read or write access restriction at a finer granularity than the table level, such as at the column level, may be obtained by defining a view of the table which omits the sensitive information. Modification (UPDATE) authority may be granted at the column level without requiring that a view be specified. Write access may be assigned to allow any combination of the following capabilities: row modification (UPDATE), row insertion (INSERT), and row deletion (DELETE). The DBA assigns access privileges by grouping users with common access needs into authorization groups. The particular read and write authorities are then assigned by the DBA to each of these groups. Users with unique access requirements may also be granted privileges directly.

HP IMAGE allows read and write access to be assigned at the data set and item level. The DBA groups all users into security classes. These classes are analogous to the authorization groups of HP SQL. The DBA specifies in the database schema which security classes have read access, which have write access, and which have both read and write access to each data item and data set. When a user opens the database, the password he gives identifies the security class to which he belongs and therefore his read and write access privileges throughout the HP IMAGE database.

Logical Transaction

A logical transaction is a series of database modifications of which either all or none must be performed to leave the database in a consistent state. The particular grouping of modifications defined by the user to be a transaction will vary depending on the particular application. An example of a transaction is an accounting entry to pay a bill: the credit to the cash account and debit to the accounts payable account must both be performed to avoid leaving the accounts out of balance. This concept of a logical transaction is essential to ensuring that data integrity is maintained when multiple users are concurrently accessing the database or in the event of a system failure.

Concurrency

ALLBASE preserves data integrity when multiple users are accessing a database, through a comprehensive locking scheme based on the transaction concept. When a user begins a transaction, a lock is automatically granted for each page read or modified by the transaction (a page is a unit of data storage which contains 4,096 characters). This ensures that no one else may update the data on those pages while the user is reading or updating them. If data is only being read, then other users are not prevented from reading it simultaneously; they just can't update it. If data is being updated, however, then data integrity is ensured by preventing the other users from reading or updating the data. When the user's transaction is completed, all acquired locks are automatically released.

Tables (data sets) may also be locked explicitly. This feature is provided to allow the programmer greater flexibility in applications where it is advantageous to lock large portions of the database. Since explicit locks reduce concurrency, they are not recommended for general use.

ALLBASE also contains sophisticated deadlock detection and correction capabilities. When a deadlock situation is detected, ALLBASE will rollback and release locks for one of the transactions, thereby preserving data integrity, and notify the application program that it has done so. This notification allows the application program to determine how to continue processing.

Recovery

ALLBASE ensures that the logical and physical integrity of the database is protected in the event of a program abort, system failure, or destruction of the media on which the database resides.

Rollback recovery is an automatically activated recovery feature which ensures that the database is always in a logically consistent state. ALLBASE logs write transactions to a log file on disc. In the event of a system failure or program abort, ALLBASE uses this log file to automatically back out any partially completed transactions.

The rollback capability may also be invoked in an HP SQL or an HP IMAGE program. This is a particularly valuable feature in an on-line application as it allows a user who has entered incorrect information to nullify the transaction before its completion.

Rollforward recovery protects the physical and logical integrity of the database against media failure. In the event of a hardware or software failure, the transactions from the log file are reapplied to a backup copy of the data to bring it up to the current state.

A simultaneously updated copy of the log files used for rollback or rollforward recovery may be kept on another disc to provide additional protection in the case of a disc failure.

Referential Integrity

The predefined relationships of HP IMAGE allow referential integrity to be maintained. Once a parent/child relationship has been established between HP IMAGE data sets, referential integrity is maintained by preventing the deletion of a parent record without first deleting all corresponding child records and by preventing the addition of a child record unless a corresponding parent record exists. For example, an HP IMAGE database could be easily defined which would prevent deletion of an order record (parent) until all of the shipments against that order (children) have been deleted and which also would prevent any shipment records from being inserted unless they are against a valid order.

ALLBASE: Interactive Data Access

ALLBASE includes two powerful query facilities which aid the application programmer and the database administrator in the creation and support of HP SQL and HP IMAGE databases. Interactive SQL (ISQL), for use with HP SQL databases, and IQUERY, for use with HP IMAGE databases, may be executed interactively (from a terminal) or in a batch mode. Output may be directed to a terminal or to any other output device (i.e., line printer, disc, tape). Frequently used ISQL and IQUERY procedures may be stored on disc for future use.

ISQL and IQUERY both honor the ALLBASE security system. When accessing the database, the user may only exercise the read/write access privileges which have been allowed him in the database definition.

A future release of ALLBASE will provide access using HP Visor, an interface which allows end users to perform ad hoc queries and generate customized reports. The HP Visor data sheet provides more information on the use and capabilities of this easy-to-use interface.

ISQL for HP SQL

ISQL is a database support tool integrated into HP SQL which features a powerful, nonprocedural query facility. ISQL allows SQL queries to be executed from a terminal, thus giving programmers and frequent users the ability to use this powerful language for ad hoc queries. Since the syntax of an SQL query is the same whether executed using ISQL or embedded in an application program, the programmer can cut application development time by using ISQL to test database queries before embedding them in applications. ISQL also includes a powerful bulk load facility which is particularly useful when creating test databases.

IQUERY for HP IMAGE

IQUERY is a database support tool included with ALLBASE for use with HP IMAGE. IQUERY features a powerful, command-driven interface designed for application programmers and database administrators. IQUERY provides extensive locating, reporting, and updating commands. These commands provide flexible data access on an ongoing basis and are particularly valuable when testing new database applications and structures.

ALLBASE: Access of Data via Programs

Both HP SQL and HP IMAGE databases may be easily accessed from application programs. A list of the high-level languages from which each of the databases may be accessed is included in the *Other ALLBASE Specifications* section at the end of this data sheet.

Preprocessors for HP SQL

HP SQL uses language preprocessors to provide easy, high-level access to the database via common programming languages. To access an HP SQL database from an application, SQL queries are embedded directly in the source code. Before compiling the application, it is preprocessed (precompiled). Preprocessing replaces the SQL queries with language specific calls and performs the query optimization.

The preprocessor approach improves performance by allowing query optimization to be performed when the application is compiled instead of at run time. At run time, HP SQL will detect if a change in the database structure has invalidated the access strategy for any of the queries and will automatically optimize those queries for the new structure.

Intrinsics for HP IMAGE

HP IMAGE databases are accessed from application programs via intrinsics, which are calls to library routines. Intrinsics provide a straightforward method of database access. This direct access approach is optimal in HP IMAGE since the data access paths are specified in the application program. Intrinsics also ensure maximum compatibility with TurboIMAGE applications.

ALLBASE: Creating and Maintaining Databases

Database Creation

As part of the database design process, the DBA must decide how many databases should be included in each ALLBASE DBEnvironment. A DBEnvironment may contain one or more databases. Since the DBEnvironment is the maximum scope for recovery, multiple databases which will be accessed via a single logical transaction should be placed in the same DBEnvironment. The DBEnvironment is also the level at which the data is backed up, therefore unrelated databases should be placed in separate DBEnvironments.

After the DBA has designed the database structure on paper, he may easily create the database. The first step in creating a database is the same for HP SQL and HP IMAGE. A DBEnvironment must be configured for the database unless it will be included with other databases in an existing DBEnvironment. The remaining step in creating an HP SQL database is to create the tables (CREATE TABLE command), indexes (CREATE INDEX command), and views (CREATE VIEW command) which comprise the database.

The DBA who is creating an HP IMAGE database creates the *schema* for the database after configuring a new DBEnvironment (if required). A schema is a text file which describes the structure of the database. The final step when creating an HP IMAGE database is to run the schema processor which creates the actual database structure from the schema file.

Database Restructuring

ALLBASE provides a full set of database restructuring capabilities. HP SQL and HP IMAGE support dynamic restructuring for commonly required structural changes. Dynamic restructuring allows users to continue to access data, except for the affected areas, during restructuring.

HP SQL provides dynamic restructuring for the following cases: expanding table capacities; altering security designations; adding columns; and adding or deleting indexes, views, and tables.

HP IMAGE provides dynamic restructuring capabilities for increasing data set capacities and for changing security designations.

Database Compatibility

HP IMAGE (ALLBASE) and TurboIMAGE/V

IMAGE applications may be developed which share common source code across the entire HP 3000 family. They may be run on the MPE V based HP 3000 using TurboIMAGE/V and in Native Mode on the 900 Series HP 3000 using TurboIMAGE/XL or HP IMAGE.

HP SQL (ALLBASE) and HP SQL/V

The HP SQL product on the MPE V based HP 3000 is upward compatible with the HP SQL interface of ALLBASE. An application may be moved from HP SQL/V to the HP SQL interface of ALLBASE with no major source code changes. Relational applications may also be developed which share a common source code across the entire HP 3000 family.

System Environment

ALLBASE/XL is supported on the 900 Series of HP 3000 systems. The HP SQL interface of ALLBASE/XL will be available on the first release of MPE XL. The HP IMAGE interface and dual access capability will be provided on future releases.

Ordering Information

The right to use ALLBASE/XL on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

ALLBASE/XL may be ordered separately as described in this section or at a discounted price as a part of Preconfigured 900 Series systems.

Product

Number	Product Description
36216A	ALLBASE/XL Right-to-Use product
36216R	ALLBASE/XL Right to Copy product with sublicense

Documentation

Part

Number	Document Description
36216-90001	SQL Reference Manual
36216-90002	HP IMAGE Reference Manual
36216-90003	IQUERY Reference Manual
36216-90004	ISQL Reference Manual
36216-90005	HP SQL Database Administration Guide
36216-90006	HP SQL COBOL Application Programming Guide
36216-90007	HP SQL Pascal Application Programming Guide
36216-90008	Migrating to ALLBASE/XL
36216-90009	HP SQL Message Manual
36216-90010	HP IMAGE Message Manual
36216-90012	ALLBASE Quick Reference Guide

Customer Training

Classes tailored to the needs of specific groups of ALLBASE users are available through the local sales office. Each course includes instruction and labs.

Contact the local sales office for course outlines and schedules.

Other ALLBASE Specifications

	HP SQL	HP IMAGE
Data Types	Decimal (packed) IEEE Standard, Double Precision Floating Point (8 byte) Integer (2 & 4 byte) Fixed Length (<3996 bytes) Character Variable Length (<3996 bytes) Character	Decimal (packed & zoned) Floating Point (4 & 8 byte) Integer (2, 4, & 8 byte) Fixed Length (<3996 bytes) Character
Languages	COBOL Pascal	COBOL Pascal FORTRAN
Passwords/Security Groups (Classes) per Database	Unlimited Groups	63 Classes
Maximum Database Parameters		
Tables (sets) per database	Unlimited	199
Records per table (set)	Unlimited	Unlimited
Record Length	3996 bytes	3996 bytes
Columns (items) per table (set)	255	255
Field (item) Length	3996 bytes	3996 bytes
Sub-items per item	n/a	255
Children per parent	n/a	31
Indices per table	Unlimited	n/a
Search Items per set	n/a	31
Columns per index	15	n/a
Sort Items per path	n/a	8

HEWLETT-PACKARD

Product Number 36215A

HP SQL/V

For MPE V Based HP 3000 Systems

A Relational Database Compatible with ALLBASE/XL

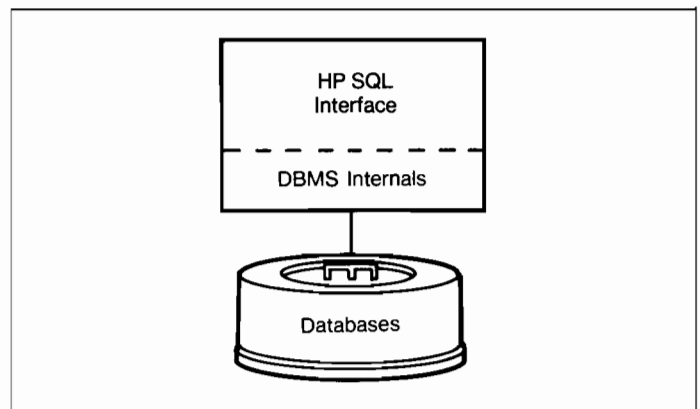
HP SQL is a functionally complete relational database management system (DBMS) which, being upward compatible with ALLBASE on the 900 Series, enables relational applications to be developed for execution across the entire HP 3000 product family.

HP SQL offers a significant programmer productivity advantage over network model and hierarchical model DBMS during application development and application maintenance. Since an HP SQL query requires that the programmer specify only what data is required, significantly less query preplanning and coding is required; HP SQL databases are easier to query on an ad hoc basis, freeing the user from specifying data access path information.

HP SQL and TurboIMAGE make a powerful team. HP SQL allows applications to be developed and put into production more quickly than with TurboIMAGE. While HP SQL has suitable performance for many production applications, TurboIMAGE is the suggested choice for applications where maximum performance is the overriding objective.

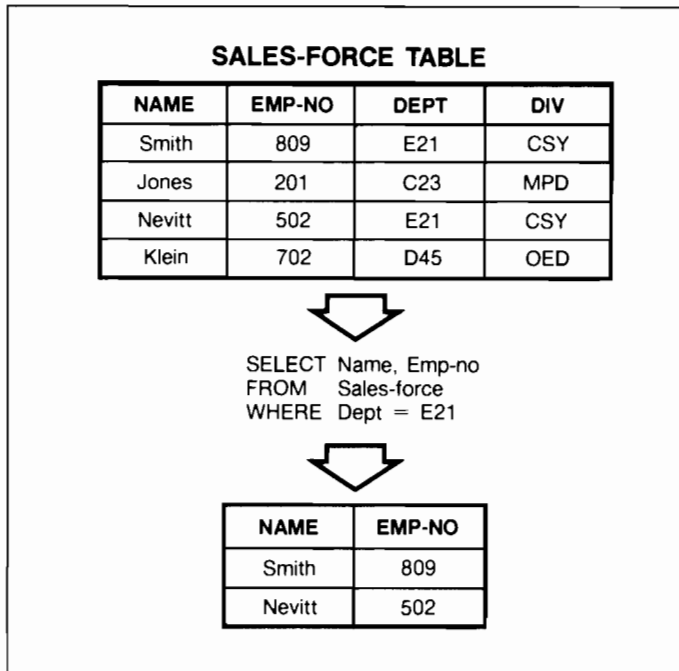
Key HP SQL Features

- Relational data model allows the user to specify only what data is required, without specifying how to retrieve it.
- SQL relational language for data definition and manipulation is the industry standard.
- Interactive SQL (ISQL) allows the database to be queried from a terminal, using the powerful SQL relational language.
- Language preprocessors allow the same statements which are used for a direct query via a terminal to be embedded in COBOL or Pascal application programs.
- Concurrent access allows multiple users to access data simultaneously.
- Security is maintained by allowing specification of appropriate levels of access privileges to individual users or groups of users.
- Data independence allows changes to be made to the database structure without requiring that applications be modified.
- Views may be defined which allow a group of users to view parts of one or more tables as a single, virtual table customized to their particular needs.
- A sophisticated query optimizer relieves the programmer of the details of query planning.
- Fast data access is supported through B-tree indexes.
- User-controlled transactions ensure that data is always in a consistent state.
- Automatic locking ensures data integrity in a multi-user environment by preventing access to data while it is being updated.
- Automatic (rollback) recovery guarantees logical data integrity in the event of a soft crash.
- The rollback capability may also be invoked in a program to allow erroneous data, usually generated in an on-line situation, to be purged before the transaction is completed.
- Rollforward recovery guarantees logical and physical data integrity in the event of a hard crash.
- Dynamic restructuring allows the data structure, table capacities, and security to be changed without unloading and loading the database.
- Null data values allow use of fields which are relevant to some, but not all, records in a table.
- Application programs are source code compatible with the HP SQL interface of ALLBASE/XL.



Functional Description

HP SQL is a functionally complete, relational model DBMS which features a full implementation of the industry standard SQL relational language. SQL is a nonprocedural language whose powerful commands operate on entire sets of data at a time, rather than on one record-at-a-time as with the procedural approach of network and hierarchical model databases.



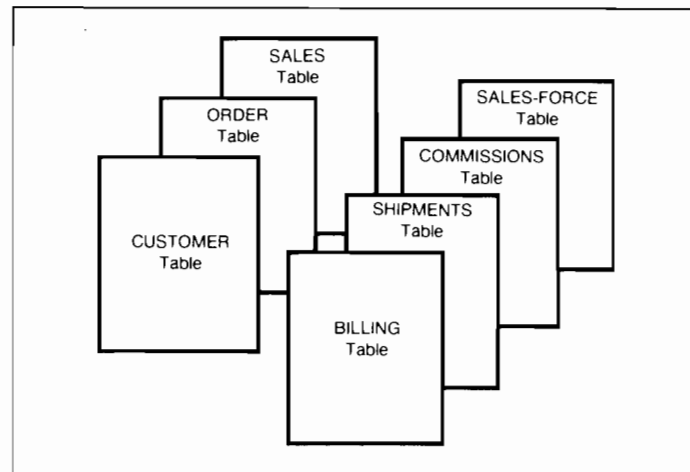
The Set-at-a-Time Query Approach of HP SQL

The select, project, and join operators of relational theory are supported. These are implemented through the SQL language, with the SELECT command for reading data. HP SQL also supports the other three data manipulation language (DML) commands of the SQL language: INSERT, for inserting groups of records; UPDATE, for updating (modifying) groups of records; and DELETE, for deleting groups of records.

HP SQL does not require that explicit relationships between different tables (data sets) be defined. Relationships are determined, at the time a query is performed, by matching values between columns common to two or more tables. Since data from any number of tables which share a common column may be related as needed, an HP SQL database consists of a group of tables which the user decides he may have a need to relate.

A sophisticated query optimizer determines the optimal data access strategy based on such factors as the presence of indexes and the relative sizes of the tables being accessed. This frees the programmer from detailed query planning without sacrificing performance.

Language preprocessors provide easy, high-level access to the database via common programming languages. This approach significantly improves performance by allowing query optimization to be performed when the application is compiled instead of at run time.



Simplified HP SQL Relational Structure

Interactive SQL (ISQL) is the comprehensive interactive interface to HP SQL. There are two distinct sets of functionality which may be used via ISQL. The first is the data definition language (DDL) which allows the database administrator (DBA) to control all aspects of database creation and modification.

The second set of ISQL accessible functionality is a complete implementation of the SQL data manipulation language (DML) which allows the programmer or frequent user to query the database from a terminal using the non-procedural SELECT, UPDATE, INSERT, and DELETE commands.

When to use HP SQL — When to use TurboIMAGE

The relational data model of HP SQL and the network data model of TurboIMAGE both have their place in meeting your business needs.

Applications may be developed in less time with HP SQL than with IMAGE. With HP SQL, the programmer must specify only what data is required, not how to access it. This significantly reduces the amount of query preplanning required during application development and maintenance. The nonprocedural nature of SQL also results in greater data independence and less application code. The high degree of data independence allows the database to be restructured without requiring that applications be modified. The reduction in code also improves the productivity of maintenance programmers by allowing the application code to be quickly understood and modified when changing business needs require that applications be updated.

The procedural TurboIMAGE DBMS allows the programmer to specify exactly how the data is to be accessed. While this requires more query preplanning, it allows the sophisticated programmer the granularity of control necessary to obtain performance approximately twice that of HP SQL/V, to optimally support throughput or response-time critical applications.

Ensuring Data Integrity

Views

A view is a table derived by defining a "filter" over one or more tables to let users of programs view only certain data in the tables. Views improve security by allowing users to access only the data they have a need to know. Since the view is not actually a physical table, use of views does not result in redundant data. When data in a table is updated, all views which use that data are automatically updated.

Security

HP SQL allows read (SELECT) access and write access privileges to be assigned at the table level. Read or write access restriction at a finer granularity than the table level, such as at the column level, may be obtained by defining a view of the table which omits the sensitive information. Modification (UPDATE) authority may be granted at the column level without requiring that a view be specified. Write access may be assigned to allow any combination of the following capabilities: row modification (UPDATE), row insertion (INSERT), and row deletion (DELETE). The DBA assigns access privileges by grouping users with common access needs into authorization groups. The particular read and write authorities are then assigned by the DBA to each of these groups. Users with unique access requirements may also be granted privileges directly.

Logical Transaction Defined

A logical transaction is a series of database modifications of which either all or none must be performed to leave the database in a consistent state. The particular grouping of modifications defined by the user to be a transaction will vary depending on the particular application. An example of a transaction is an accounting entry to pay a bill: the credit to the cash account and debit to the accounts payable account must both be performed to avoid leaving the accounts out of balance. This concept of a logical transaction is essential to ensuring that data integrity is maintained when multiple users are concurrently accessing the database or in the event of a system failure.

Concurrency

Through a comprehensive locking scheme based on the transaction concept, HP SQL preserves data integrity when multiple users are accessing a database. When a user begins a transaction, a lock is automatically granted for each page read or modified by the transaction (a page is a unit of data storage which contains 4,096 characters). This ensures that no one else may update the data on those pages while the user is reading or updating them. If data is only being read, then other users are not prevented from reading it simultaneously; they just can't update it. If data is being updated, however, then data integrity is ensured by preventing the other users from reading or updating the data. When the user's transaction is completed, all acquired locks are automatically released.

Tables may also be locked explicitly. This feature is provided to allow the programmer greater flexibility in applications where it is advantageous to lock large portions of the database. Since explicit locks reduce concurrency, they are not recommended for general use.

HP SQL also contains sophisticated deadlock detection and correction capabilities. When a lock conflict is detected, HP SQL will roll back one of the transactions, thereby preserving data integrity, and notify the application program that it has done so. This notification allows the application program to determine how to continue processing.

Recovery

Logical and physical integrity of the database is protected in the event of a program abort, system failure, or destruction of the media on which the database resides.

Rollback recovery is an automatically activated recovery feature which ensures that the database is always in a logically consistent state. HP SQL logs before-images and after-images for each write transaction to a log file on disc. In the event of a system failure or program abort, this log file is used to automatically back out any partially completed transactions.

The rollback capability may also be invoked in a program. This is a particularly valuable feature in an on-line application as it allows a user who has entered incorrect information to nullify the transaction before its completion.

Rollforward recovery protects the physical and logical integrity of the database against media failure. In the event of a hardware or software failure, the transactions from the log file are reapplied to a backup copy of the database to bring it up to the current state.

A simultaneously updated copy of the log files used for rollback or rollforward recovery may be kept on another disc to provide additional protection in the case of a disc failure.

Interactive Data Access

Integrated into HP SQL is ISQL, a database support tool which features a powerful, nonprocedural query facility. ISQL allows SQL queries to be executed from a terminal, thus giving programmers and frequent users the ability to use this powerful language for ad hoc queries. Since the syntax of an SQL query is the same whether executed using ISQL or embedded in an application program, the programmer can cut application development time by using ISQL to test database queries before embedding them in applications. ISQL also includes a powerful bulk load facility which is particularly useful when creating test databases.

ISQL honors the HP SQL security system. When accessing the database, the user may only exercise the read/write access privileges which have been allowed him in the database definition.

ISQL queries may be executed interactively (from a terminal) or in a batch mode. Output may be directed to a terminal or to any other output device (i.e., line printer, disc, tape). Frequently used ISQL procedures may be stored on disc for future use.

HP SQL databases can also be accessed using HP Visor, an interface which allows end users to perform ad hoc queries and generate customized reports. The HP Visor data sheet provides more information on the use and capabilities of this easy-to-use interface.

Access of Data via Programs

HP SQL uses language preprocessors to provide easy, high level access to the database via popular programming languages. To access a database from an application, SQL queries are embedded directly in the COBOL or Pascal source code. Before compiling the application, it is pre-processed (precompiled). Preprocessing replaces the SQL queries with language specific calls and performs the query optimization.

The preprocessor approach improves performance by allowing query optimization to be performed when the application is compiled instead of at run time. At run time, HP SQL will detect if a change in the database structure has invalidated the access strategy for any of the queries and will automatically optimize those queries for the new structure.

Creating and Maintaining Databases

Database Creation

As part of the database design process, the DBA must decide how many databases should be included in each HP SQL DBEnvironment. A DBEnvironment may contain one or more databases. Since the DBEnvironment is the maximum scope for recovery, multiple databases which will be accessed via a single logical transaction should be placed in the same DBEnvironment. The DBEnvironment is also the granularity at which the data is backed up, therefore unrelated databases should be placed in separate DBEnvironments.

After the DBA has designed the database structure on paper, he may easily create the database. First, a DBEnvironment must be configured for the database unless it will be included with other databases in an existing DBEnvironment. The DBA then needs to create the tables (CREATE TABLE command), indexes (CREATE INDEX command), and views (CREATE VIEW command) which comprise the database.

Databases can be designed, modified and tuned on a development system, and then easily transferred to a production system using the SQLGEN utility.

Dynamic Restructuring

Dynamic restructuring allows users to continue to access data, except for the affected areas, during restructuring. HP SQL's powerful dynamic restructuring capabilities include expanding table capacities; altering security designations; adding columns; and adding or deleting indexes, views, and tables.

Compatibility with the HP SQL Interface of ALLBASE/XL

HP SQL/V on the MPE V based HP 3000 is upward compatible with the HP SQL interface of ALLBASE/XL. An application may be moved from HP SQL/V to the HP SQL interface of ALLBASE/XL with no major source code changes. This common interface also enables relational applications to be developed which share common source code across the entire HP 3000 family.

System Environment

HP SQL/V is supported on MPE V based HP 3000 systems (T-delta 4-MIT or later). HP SQL/V is NOT supported in Compatibility Mode on the 900 Series of HP 3000 systems.

Minimum Configuration

4 MByte of memory, 2 Discs

Typical Configuration

8 MByte of memory, 3 Discs

HP SQL Specifications

Data Types

- Decimal (packed)
- Double Precision Floating Point (8 byte)
- Integer (2 and 4 byte)
- Fixed Length (< 3996 bytes) Character
- Variable Length (< 3996 bytes) Character

Languages

COBOL and Pascal

Passwords/Security Groups per Database

Unlimited

Maximum Database Parameters

Tables per database	Unlimited
Records per table	Unlimited
Record length	3996 bytes
Columns per table	64
Field length	3996 bytes
Indexes per table	Unlimited
Columns per index	15

Ordering Information

The right to use HP SQL/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
36215A	HP SQL/V Right-to-Use product
36215R	HP SQL/V Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
36215-90001	HP SQL/V SQL Reference Manual
36215-90002	HP SQL/V ISQL Reference Manual
36215-90003	HP SQL/V Database Administration Guide
36215-90004	HP SQL/V COBOL Application Programming Guide
36215-90005	HP SQL/V Message Manual
36215-90006	HP SQL/V Quick Reference Guide
36215-90007	HP SQL/V Pascal Application Programming Guide

Customer Training

Three classes, each one tailored to the needs of a specific group of HP SQL users, are available through the local sales office. Each course includes instruction and labs.

Course Number	Course Name
36207A	HP SQL/V Application Programming Course
36225A	HP SQL/V Database Administration Course
36229A	HP SQL/V Introductory Interactive SQL User's Course

Contact the local sales office for course outlines and schedules.

HEWLETT-PACKARD

Product Number 30391A

TurboIMAGE/XL

For the 900 Series of HP 3000 Systems

Network Model DBMS

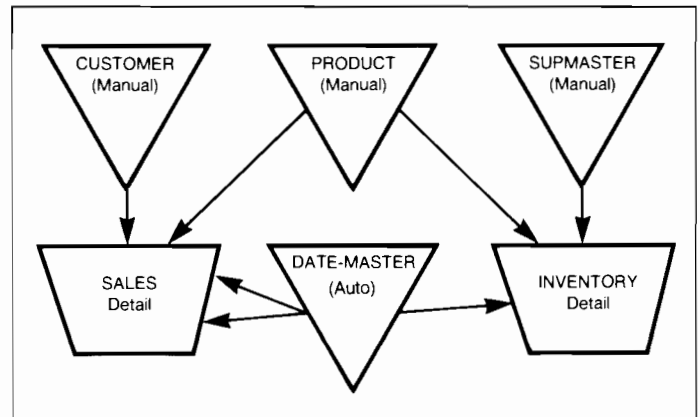
TurboIMAGE/XL continues in the DATAPRO award-winning tradition of IMAGE/3000 and TurboIMAGE/V database management systems (DBMS) designed for use with the HP 3000 computer family. TurboIMAGE/XL offers Native Mode operation of TurboIMAGE with compatible functionality and Native Mode performance. The database architecture is a two-level network model structure with owner/member data set relationships. This general purpose database system is included with the Fundamental Operating Software (FOS) on pre-configured 900 Series HP 3000 computer systems.

TurboIMAGE offers the capabilities to describe data structures, define data relationships, and create databases. A high level language interface is provided for access and maintenance of the data within the database. This interface supports applications written in COBOL, RPG, Pascal, SPL, FORTRAN, BASIC, and Transact.

TurboIMAGE allows data to be related logically between data sets. This linking minimizes data redundancy and facilitates fast information retrieval. In addition, a two-level security facility (at the data set and data item level) protects the database against unauthorized access.

Since locking is controllable from the database level to the data entry level, applications and/or interactive users may share the database concurrently. With NS/3000 networking products, programs are able to access databases remotely. Remote access allows a user on one HP 3000 to access a TurboIMAGE database on another HP 3000 computer.

Performance enhancements have been incorporated into TurboIMAGE that will support a wide range of database needs including applications requiring a large and high-performance database as well as applications with database requirements less demanding. The TurboIMAGE database management system provides the basis for developing information systems tailored to today's corporate, industrial, and educational needs.



Simplified TurboIMAGE Network Structure

Terminology

The brief glossary that follows will clarify and provide familiarity with common database terms which will be used.

Data item — smallest accessible unit of data, may consist of a field or a logically contiguous group of fields.

Data entry — groups of data items composing a record.

Data set — a collection of data entries where each entry contains values for the same data items.

Privacy — ability to restrict access to information.

Security — controlling feature to allocate privacy restriction at the data set and data item level.

Transaction — a logical unit of work defined by the programmer for the purpose of logging and recovery.

Concurrency — the ability to allow users (either batch and/or interactive) to share data in the database simultaneously.

Intrinsic — a library routine callable from application programs to perform specific functions.

Account	Last Name	First Name	Initial	Street Address	City	State	Zip	Credit Rating
12345678	Miller	James	L	1645 Marshall Avenue	Glendale	Az	85301	
95430301	Brighton	Abigail	S	72 E Hampton Drive	Carmel	Ca	93921	
54777833	Graziano	Isabel	M	113 Shasta Lane	Santa Clara	Ca	95050	

Data Items and Data Entries within a Data Set

Key TurboIMAGE Features

- Logically related files can be handled as a single entity (a database), relieving the application of low-level details.
- Network structure allows fast access to complex relationships among data.
- TurboIMAGE intrinsics callable from COBOL, RPG, Pascal, SPL, FORTRAN, BASIC, and Transact provide a high-level programmatic interface.
- Files within a database may stand alone or be logically linked together, allowing flexible data storage.
- Serial, direct, calculated, and chained access methods are available, to provide flexible choices for data retrieval.
- Access to multiple databases allows you to logically associate the data while maintaining database independence.
- Deleted record space is automatically reused to enhance efficient access.
- Concurrent interactive and/or batch access allows multiple users to access data simultaneously.
- Privacy and security allows the control to restrict certain information from unauthorized users. Two-level security at the data set and data item levels, defines subsets of the database providing customizable access to data.
- Transaction Manager guarantees the physical integrity of the database in the event of a soft crash. This is the default operating mode of TurboIMAGE/XL.
- Roll-back recovery guarantees the logical and physical integrity of the database in the event of a soft crash.
- Roll-forward recovery guarantees the logical and physical integrity in the event of a hard crash.
- Changelog Facility provides a logical link to new log files. It automatically switches to the new log files on disc or tape when the current log file becomes full.
- User defined locking strategy, at database, data set, and data entry level allows optimum concurrency level control.

When to Use TurboIMAGE — When to Use HP SQL/XL

The network data model of TurboIMAGE and the relational data model of HP SQL/XL both have their place in meeting your business needs. HP SQL/XL is HP's relational DBMS as part of ALLBASE.

TurboIMAGE is well suited for high-volume inquiry and update transaction oriented applications. Characteristic of such applications is that they will be using predefined and stable database structures in a repetitive manner. Performance considerations and detailed tuning facilities are typically more important than high-level access to the data. The procedural TurboIMAGE DBMS allows the programmer to specify exactly how the data is to be accessed. While this requires more query preplanning, it allows the sophisticated programmer the granularity of control necessary to attain optimum performance in throughput for response-time critical applications.

Applications may be developed in less time with HP SQL than with TurboIMAGE. With HP SQL, the programmer must specify only what data is required, not how to access it. Therefore in applications where programmer productivity is important and speed of data access is non-critical, HP SQL should be considered strongly.

TurboIMAGE Components

TurboIMAGE consists of three components:

Data Definition Language. A free-format data definition language (DDL) is provided with TurboIMAGE. Through the DDL, the data base administrator defines the database by describing the items (symbolic name, data type, and length), security, data sets, data set relationships, and size. This database description file is called the *schema*. The schema is then used by a utility that creates the database structure including the data set file space.

Data Manipulation Language. A set of high-level, callable TurboIMAGE intrinsics comprise the data manipulation language (DML). These intrinsics support programmatic access to or maintenance of the data within the TurboIMAGE database. By supplying information such as the database name, the data set name, and the type of access desired, the data base manipulation language locates the data and performs the requested function. The application program simply "calls" the desired intrinsic. The TurboIMAGE DML includes the following intrinsics:

DBOPEN initiates access to a database.

DBCLOSE terminates access to a database or a data set, or rewinds a data set.

DBLOCK temporarily provides exclusive control of a database, one or more data sets, or one or more data entries.

DBUNLOCK relinquishes the locks acquired by all previous calls to DBLOCK.

DBBEGIN designates the beginning of a logical TurboIMAGE transaction.

DBEND designates the end of a TurboIMAGE transaction.

DBMEMO writes a user-informational message to the TurboIMAGE log file.

DBFIND prepares for chained access to data entries.

DBGET retrieves data entries from a data set.

DBPUT adds new data entries to a data set.

DBUPDATE modifies existing data entries.

DBDELETE deletes data entries from a data set.

DBINFO returns information about the database currently being accessed.

DBCONTROL enables or disables the option to defer output.

DBERROR moves error message to a buffer specified by the calling program.

DBEXPLAIN prints message on the \$STDLIST device.

Database Utilities. Stand-alone utility programs aid in the creation of data sets and the maintenance of a database. These utilities may be run in batch or interactive mode.

DBUTIL is a general purpose utility that performs several different maintenance and administrative functions. Some of the functions of DBUTIL include:

- Allocate and initialize disc space for a database.
- Re-initialize all the data sets of a database.
- Purge the root file and all of the data sets within the database.
- Activate/deactivate the database for Remote Database Access.
- Enable/disable the database for logging and Roll-forward recovery.
- Enable/disable the database for Roll-back recovery.
- Reassign database passwords.
- Display the status of locks currently obtained.
- Move TurboIMAGE file from one device to another.
- Change file system security for TurboIMAGE database files.

DBSTORE transfers a physical copy of a database to magnetic tape or serial disc.

DBRESTOR copies a database from magnetic tape or serial disc back to disc.

DBUNLOAD produces a logical copy (data only) of a database on magnetic tape or serial disc.

DBLOAD loads data from a magnetic tape or serial disc (created with DBUNLOAD) into an existing empty database on disc.

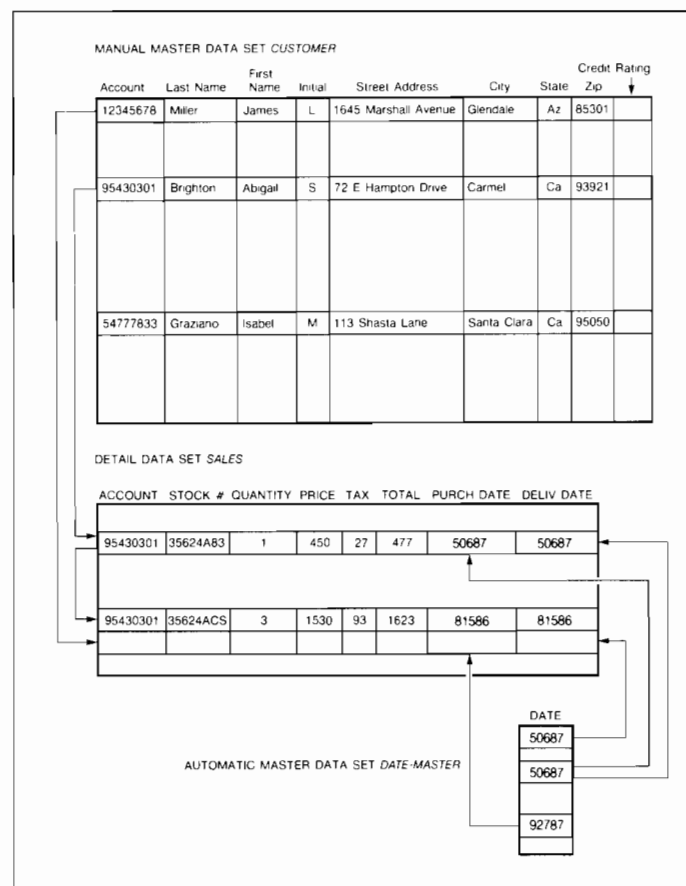
DBRECOV reads the log file to recover a database in the event of a system failure, with roll-back or roll-forward recovery.

Data Types

TurboIMAGE supports signed binary integer (single and double word), unsigned integer, binary, real, double precision real, ASCII character strings, packed decimal, and zoned decimal data types.

Data Set Types

TurboIMAGE supports two types of data sets, master and detail. Master data sets are used to keep information relating to a unique entity; for example, information describing a customer. They allow for rapid retrieval of a data entry, since one of the items in the entry is a search item (a key). A search item serves as an index into detail data sets. Through the search item, a master data set may be linked to up to 16 detail data sets. Detail data sets are used to record information about related events; for example, all of the purchases made by a particular customer. A detail data set may be defined with up to 16 search items and therefore be linked to a maximum of 16 master data sets. Items in detail data sets with identical key values are maintained in an internal linked-list called a chain. TurboIMAGE/XL will allow a chain length to be equal to the number of data entries in its respective data set.



Master and Detail Data Set Relationships

In addition to data sets that are linked together, TurboIMAGE allows both Master and Detail data sets to "stand-alone". This means that there are no links to other data sets within the database. Calculated access is possible with a stand-alone master data set. TurboIMAGE will allow up to 199 data sets to be defined within a single database. These data sets may be in any combination of master/detail relationships.

Access

TurboIMAGE supports serial (chronological), direct, calculated (hashed), and chained access. Access to data entries in a master data set may be hashed (based on the key value of the data entry), or serially read (forward or backward). Access to a data entry in a detail data set is usually through a particular key using an argument value (e.g., name = SMITH). This type of access is called chained. Chained access may be done in either read-previous or read-next modes. It is also possible to access data in a detail data set via direct (by relative record number) or serial (forward or backward) modes. Access methods may be chosen to suit the particular application or subroutine.

Security Facility

Logical views are maintained within TurboIMAGE. The logical view security is implemented through a class-type privacy and security scheme. The database administrator associates a password with each user class to be defined in the database. The user classes are then granted "read" or "read-and-write" access to any or all of the data items and data sets. End users must supply their password when accessing the database. That password will provide end users access to their logical view of the data base. TurboIMAGE allows up to 63 defined passwords within a single database.

Database Integrity

Database Locking

Locking is possible on three levels: the database, data set, and data entries (by locking the value of a data item). TurboIMAGE will allow mixed levels of locking where one user may request a lock on a data set while a second user may request a lock on a data entry. Locking at the data entry level will provide the most concurrency. Locking may be done unconditionally where the application waits for a successful lock, or conditionally where the application can perform other things while a lock is held by other applications.

Recovery

TurboIMAGE/XL recovery options have been modified to operate most effectively with the MPE XL operating system. TurboIMAGE/XL provides default recovery as well as three other methods of recovery: Intrinsic Level Recovery (ILR), Rollforward recovery, and Rollback recovery. In default recovery mode, TurboIMAGE/XL ensures the physical integrity of the database. TurboIMAGE intrinsics that alter the chains in the database are logged to the Transaction Manager (XM) log file (see MPE XL Operating System data sheet). With default mode, these logged intrinsics may not be immediately written to disc. If ILR is enabled the log records are flushed to disc immediately. If a hardware or software failure occurs before the intrinsic completes, the MPE XL operating system using Transaction Manager will back out the interrupted intrinsic.

Roll-forward recovery ensures the logical and physical integrity of the database. TurboIMAGE transactions are logged automatically to a tape or disc transaction oriented log file. In the event of a hardware or software failure, the recovery system reads the log file and re-executes those transactions which were successfully completed, after the back-up database has been restored.

Roll-back recovery ensure the logical and physical integrity of the database. As with Roll-forward, TurboIMAGE transactions are logged automatically to a tape of disc log file. The user log file and the database must reside in the same volume set. In the event of a hardware or software failure, the recovery system will "roll-back" any incomplete transactions. Roll-back recovery utilizes ILR, to provide database integrity.

Restructuring

DBLOAD and DBUNLOAD utilities provide for restructuring databases in a number of ways, including:

- Changing data item name or data set name
- Increasing or decreasing data set capacities
- Adding or removing data items at the end of a data entry
- Changing data set relationships

Accessing the Database

A TurboIMAGE database can be accessed in the following ways:

- From COBOL, FORTRAN, and BASIC through a "CALL" statement to the TurboIMAGE intrinsics.
- From RPG through the chain and read statements after the database has been declared in the file specifications section.
- From SPL and Pascal through intrinsic calls to TurboIMAGE.
- From the QUERY language function.
- From the Transact interface function.

QUERY/V Inquiry Facility

QUERY/V is an inquiry facility designed for use by application programmers, database designers, and database administrators. Application programmers and database designers can use QUERY as a tool to validate the database structure before coding has begun. Using the query commands, the programmer can perform module and system testing against the database structure. In addition, QUERY can be used as a way to enter data into a database structure in order to test the logic of the application program.

A database administrator or database designer can use QUERY as a tool for maintaining TurboIMAGE/XL databases. The ADD, DELETE, and REPLACE commands, designed for this purpose, allow data item value modification without writing programs. For additional information on QUERY, refer to the QUERY/V data sheet.

Remote Database Access

NS/3000 permits manipulation of a TurboIMAGE database residing on a remote system. The same security provisions apply to remote access as apply to local access. The user must log on to the system(s) and provide passwords to TurboIMAGE.

To isolate both the user and the application program, TurboIMAGE allows the database administrator to create a directory called a database-access file. This access file defines the location of the database and the authorized users. Only the administrator need be concerned with actual locations; TurboIMAGE and NS/3000 handle the rest. This flexibility allows the administrator to relocate databases without affecting the users' operating procedures or changing any application programs.

TurboIMAGE Specifications

- Data item names per database: 1023
- Data items per data entry: 255
- Data sets per database: 199
- Detail data sets per master set: 16
- Master data sets per detail set: 16
- Search items (keys) per detail data set: 16
- Maximum entry size: 4096 bytes
- Entries per data set: 2 billion (with an assumed blocking factor of 255)
- Entries per chain: 2 billion
- Characters per database name: 6
- Characters per password: 8
- Characters per data set name: 16
- Characters per data item name: 16

Database Compatibility

TurboIMAGE applications may be run in Native Mode on the 900 Series of HP 3000 systems or in Compatibility Mode. IMAGE applications may therefore be developed which share a common source code across the entire HP 3000 family.

IMAGE is also available on the HP 1000 family and selected models of the HP 9000 family. Although very similar in nature, their external specifications and internal storage format is different. The reference manuals for the individual IMAGE products indicate the features and specifications supported by the particular IMAGE model. It is possible to move data from one computer to another, if the differences between the IMAGE databases are taken into consideration in the design phase.

System Environment

TurboIMAGE/XL is supported on the 900 Series of HP 3000 systems.

Ordering Information

The right to use TurboIMAGE/XL on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

TurboIMAGE/XL is included in the Fundamental Operating Software on the 900 Series of HP 3000 systems.

Documentation

Part Number	Document Description
30391-90001	TurboIMAGE Reference Manual
32215-90042	QUERY/V Reference Manual

Customer Training

A one week class on TurboIMAGE and QUERY/V is available through the local sales office. The course includes instruction and labs. Training is provided on:

- Creating a TurboIMAGE database using the Data Definition Language
- Practice with the Data Manipulation Language
- Using QUERY against a TurboIMAGE database

Contact the local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Number 32215C

TurboIMAGE/V

For MPE V Based HP 3000 Systems

Network Model DBMS

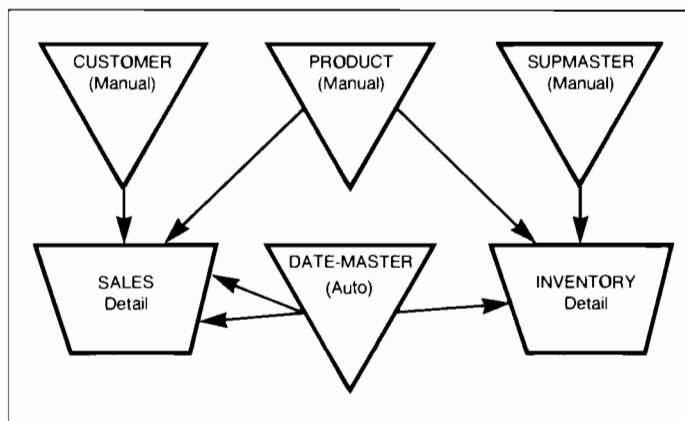
IMAGE/3000 is a DATAPRO award-winning database management system (DBMS) designed for use with the HP 3000 computer family. TurboIMAGE/V offers the same award-winning features provided by IMAGE/3000 plus increased performance and functionality. Replacing IMAGE/3000, this general purpose database system is included with the Fundamental Operating Software (FOS) on MPE V based HP 3000 computer systems. The database architecture is a two-level network model structure with owner/member data set relationships.

TurboIMAGE offers the capabilities to describe data structures, define data relationships, and create databases. A high-level language interface is provided for access and maintenance of the data within the database. This interface supports applications written in COBOL, RPG, Pascal, SPL, FORTRAN, BASIC, and Transact.

TurboIMAGE allows data to be related logically between data sets. This linking minimizes data redundancy and facilitates fast information retrieval. In addition, a two-level security facility (at the data set and data item level) protects the database against unauthorized access.

Since locking is controllable from the database level to the data entry level, applications and/or interactive users may share the database concurrently. With DS/3000 or NS/3000 networking products, programs are able to access databases remotely. Remote access allows a user on one HP 3000 to access a TurboIMAGE database on another HP 3000 computer.

Performance enhancements have been incorporated into TurboIMAGE that will support a wide range of database needs including applications requiring a large and high-performance database as well as applications with database requirements less demanding. The TurboIMAGE database management system provides the basis for developing information systems tailored to today's corporate, industrial, and educational needs.



Simplified TurboIMAGE Network Structure

Terminology

The brief glossary that follows will clarify and provide familiarity with common database terms which will be used.

Data item — smallest accessible unit of data, may consist of a field or a logically contiguous group of fields.

Data entry — groups of data items composing a record.

Data set — a collection of data entries where each entry contains values for the same data items.

Privacy — ability to restrict access to information.

Transaction — a logical unit of work defined by the programmer for the purpose of logging and recovery.

Concurrency — the ability to allow users (either batch and/or interactive) to share data in the database simultaneously.

Intrinsic — a library routine callable from application programs to perform specific functions.



Account	Last Name	First Name	Initial	Street Address	City	State	Zip	Credit Rating
12345678	Miller	James	L	1645 Marshall Avenue	Glendale	Az	85301	
95430301	Brighton	Abigail	S	72 E Hampton Drive	Carmel	Ca	93921	
54777833	Graziano	Isabel	M	113 Shasta Lane	Santa Clara	Ca	95050	

Data Items and Data Entries within a Data Set

Key TurboIMAGE Features

- Logically related files can be handled as a single entity (a database), relieving the application of low-level details.
- Network structure allows fast access to complex relationships among data.
- TurboIMAGE intrinsics callable from COBOL, RPG, Pascal, SPL, FORTRAN, BASIC, and Transact provide a high-level programmatic interface.
- Files within a database may stand alone or be logically linked together, allowing flexible data storage.
- Serial, direct, calculated, and chained access methods are available, to provide flexible choices for data retrieval.
- Access to multiple databases allows you to logically associate the data while maintaining database independence.
- Deleted record space is automatically reused to enhance efficient access.
- Concurrent interactive and/or batch access allows multiple users to access data simultaneously.
- Privacy and security allows the control to restrict certain information from unauthorized users. Two-level security at the data set and data item levels, defines subsets of the database providing customizable access to data.
- Intrinsic Level Recovery guarantees the physical integrity of the database in the event of a soft crash. ILR can be enabled through DBUTIL.
- Roll-back recovery guarantees the logical and physical integrity of the database in the event of a soft crash.
- Roll-forward recovery guarantees the logical and physical integrity in the event of a hard crash.
- Changelog Facility provides a logical link to new log files. It automatically switches to the new log files on disc or tape when the current log file becomes full.
- User defined locking strategy, at database, data set, and data entry level allows optimum concurrency level control.

When to Use TurboIMAGE — When to Use HP SQL/V

The network data model of TurboIMAGE and the relational data model of HP SQL/V both have their place in meeting your business needs. HP SQL/V is HP's relational DBMS on MPE V based systems.

TurboIMAGE is well suited for high-volume inquiry and update transaction oriented applications. Characteristic of such applications is that they will be using predefined and stable database structures in a repetitive manner. Performance considerations and detailed tuning facilities are typically more important than high-level access to the data. The procedural TurboIMAGE DBMS allows the programmer to specify exactly how the data is to be accessed. While this requires more query preplanning, it allows the sophisticated programmer the granularity of control necessary to attain optimum performance in throughput for response-time critical applications.

Applications may be developed in less time with HP SQL than with TurboIMAGE. With HP SQL, the programmer must specify only what data is required, not how to access it. Therefore in applications where programmer productivity is important and speed of data access is non-critical, HP SQL should be considered strongly.

TurboIMAGE Components

TurboIMAGE consists of three components:

Data Definition Language. A free-format data definition language (DDL) is provided with TurboIMAGE. Through the DDL, the database administrator defines the database by describing the items (symbolic name, data type, and length), security, data sets, data set relationships, and size. This database description file is called the *schema*. The schema is then used by a utility that creates the database structure including the data set file space.

Data Manipulation Language. A set of high-level, callable TurboIMAGE intrinsics comprise the data manipulation language (DML). These intrinsics support programmatic access to or maintenance of the data within the TurboIMAGE database. By supplying information such as the database name, the data set name, and the type of access desired, the data base manipulation language locates the data and performs the requested function. The application program simply "calls" the desired intrinsic. The TurboIMAGE DML includes the following intrinsics:

DBOPEN initiates access to a database.

DBCLOSE terminates access to a database or a data set, or rewinds a data set.

DBLOCK temporarily provides exclusive control of a database, one or more data sets, or one or more data entries.

DBUNLOCK relinquishes the locks acquired by all previous calls to DBLOCK.

DBBEGIN designates the beginning of a logical TurboIMAGE transaction.

DBEND designates the end of a TurboIMAGE transaction.

DBMEMO writes a user-informational message to the TurboIMAGE log file.

DBFIND prepares for chained access to data entries.

DBGET retrieves data entries from a data set.

DBPUT adds new data entries to a data set.

DBUPDATE modifies existing data entries.

DBDELETE deletes data entries from a data set.

DBINFO returns information about the database currently being accessed.

DBCONTROL enables or disables the option to defer output.

DBERROR moves error message to a buffer specified by the calling program.

DBEXPLAIN prints message on the \$STDLIST device.

Database Utilities. Stand-alone utility programs aid in the creation of data sets and the maintenance of a database. These utilities may be run in batch or interactive mode.

DBUTIL is a general purpose utility that performs several different maintenance and administrative functions. Some of the functions of DBUTIL include:

- Allocate and initialize disc space for a database.
- Re-initialize all the data sets of a database.
- Purge the root file and all of the data sets within the database.
- Activate/deactivate the database for Remote Database Access.
- Enable/disable the database for logging and Roll-forward recovery.
- Enable/disable the database for Roll-back recovery.
- Reassign database passwords.
- Display the status of locks currently obtained.
- Move TurboIMAGE file from one device to another.
- Change file system security for TurboIMAGE database files.

DBSTORE transfers a physical copy of a database to magnetic tape or serial disc.

DBRESTOR copies a database from magnetic tape or serial disc back to disc.

DBUNLOAD produces a logical copy (data only) of a database on magnetic tape or serial disc.

DBLOAD loads data from a magnetic tape or serial disc (created with DBUNLOAD) into an existing empty database on disc.

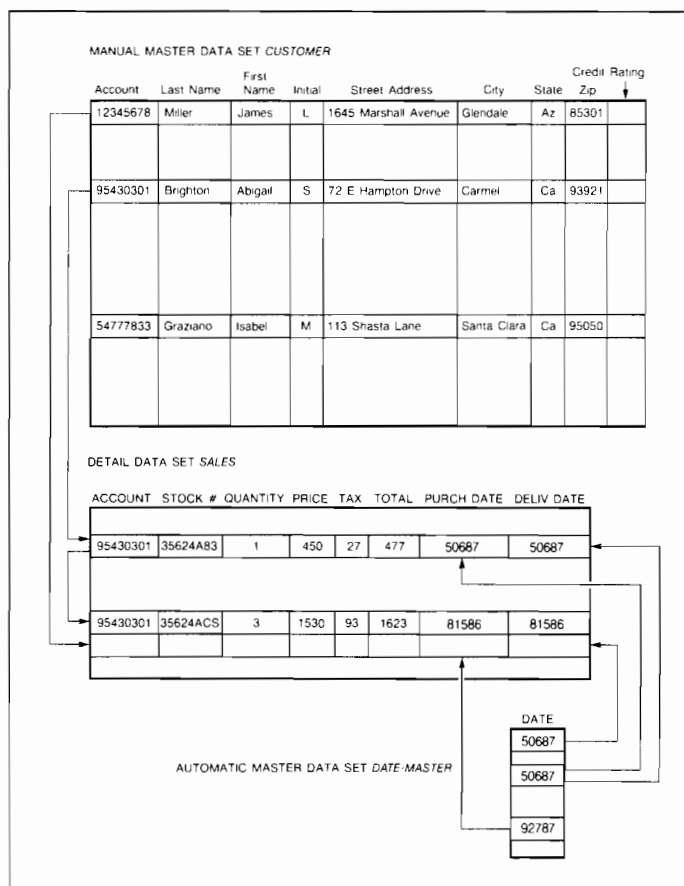
DBRECOV reads the log file to recover a database in the event of a system failure, with roll-back or roll-forward recovery.

Data Types

TurboIMAGE supports signed binary integer (single and double word), unsigned integer, binary, real, double precision real, ASCII character strings, packed decimal, and zoned decimal data types.

Data Set Types

TurboIMAGE supports two types of data sets, master and detail. Master data sets are used to keep information relating to a unique entity; for example, information describing a customer. They allow for rapid retrieval of a data entry, since one of the items in the entry is a search item (a key). A search item serves as an index into detail data sets. Through the search item, a master data set may be linked to up to 16 detail data sets. Detail data sets are used to record information about related events; for example, all of the purchases made by a particular customer. A detail data set may be defined with up to 16 search items and therefore be linked to a maximum of 16 master data sets. Items in detail data sets with identical key values are maintained in an internal linked-list called a chain. TurboIMAGE/V will allow a chain length to be equal to the number of data entries in its respective data set.



Master and Detail Data Set Relationships

In addition to data sets that are linked together, TurboIMAGE allows both Master and Detail data sets to "stand-alone". This means that there are no links to other data sets within the database. Calculated access is possible with a stand-alone master data set. TurboIMAGE will allow up to 199 data sets to be defined within a single database. These data sets may be in any combination of master/detail relationships.

Access

TurboIMAGE supports serial (chronological), direct, calculated (hashed), and chained access. Access to data entries in a master data set may be hashed (based on the key value of the data entry), or serially read (forward or backward). Access to a data entry in a detail data set is usually through a particular key using an argument value (e.g., name = SMITH). This type of access is called chained. Chained access may be done in either read-previous or read-next modes. It is also possible to access data in a detail data set via direct (by relative record number) or serial (forward or backward) modes. Access methods may be chosen to suit the particular application or subroutine.

Security Facility

Logical views are maintained within TurboIMAGE. The logical view security is implemented through a class-type privacy and security scheme. The database administrator associates a password with each user class to be defined in the database. The user classes are then granted "read" or "read-and-write" access to any or all of the data items and data sets. End users must supply their password when accessing the database. That password will provide end users access to their logical view of the data base. TurboIMAGE allows up to 63 defined passwords within a single database.

Database Integrity

Database Locking

Locking is possible on three levels: the database, data set, and data entries (by locking the value of a data item). TurboIMAGE will allow mixed levels of locking where one user may request a lock on a data set while a second user may request a lock on a data entry. Locking at the data entry level will provide the most concurrency. Locking may be done unconditionally where the application waits for a successful lock, or conditionally where the application can perform other things while a lock is held by other applications.

Recovery

TurboIMAGE offers three methods of recovery: Intrinsic Level Recovery (ILR), Roll-forward recovery, and Roll-back recovery. With Intrinsic Level Recovery, TurboIMAGE ensures the physical integrity of the database. TurboIMAGE/V intrinsics that alter the chains in the database are logged to an ILR log file. Once the intrinsic completes, the ILR log file is re-initialized. If a hardware or software failure occurs before the intrinsic completes, TurboIMAGE will reapply the interrupted intrinsic.

Roll-forward recovery ensures the logical and physical integrity of the database. TurboIMAGE transactions are logged automatically to a tape or disc log file. This log file is transaction oriented as opposed to the intrinsic oriented log file used by ILR. In the event of a hardware or software failure, the recovery system reads the log file and re-executes those transactions which were successfully completed, after the back-up database has been restored.

Roll-back recovery ensures the logical and physical integrity of the database. As with Roll-forward, TurboIMAGE transactions are logged automatically to a tape or disc log file. In the event of a hardware or software failure, the recovery system will "roll-back" any incomplete transactions. Roll-back recovery utilizes ILR, to provide database integrity.

Restructuring

DBLOAD and DBUNLOAD utilities provide for restructuring databases in a number of ways, including:

- Changing data item name or data set name
- Increasing or decreasing data set capacities
- Adding or removing data items at the end of a data entry
- Changing data set relationships

Accessing the Database

A TurboIMAGE database can be accessed in the following ways:

- From COBOL, FORTRAN, and BASIC through a "CALL" statement to the TurboIMAGE intrinsics.
- From RPG through the chain and read statements after the database has been declared in the file specifications section.
- From SPL and Pascal through intrinsic calls to TurboIMAGE.
- From the QUERY language function.
- From the Transact interface function.

QUERY/V Inquiry Facility

QUERY/V is an inquiry facility designed for use by application programmers, database designers, and database administrators. Application programmers and database designers can use QUERY as a tool to validate the database structure before coding has begun. Using the query commands, the programmer can perform module and system testing against the database structure. In addition, QUERY can be used as a way to enter data into a database structure in order to test the logic of the application program.

A database administrator or database designer can use QUERY as a tool for maintaining TurboIMAGE/V databases. The ADD, DELETE, and REPLACE commands, designed for this purpose, allow data item value modification without writing programs. For additional information on QUERY, refer to the QUERY/V data sheet.

Remote Database Access

NS/3000 or DS/3000 permits manipulation of a TurboIMAGE database residing on a remote system. The same security provisions apply to remote access as apply to local access. The user must log on to the system(s) and provide passwords to TurboIMAGE.

To isolate both the user and the application program, TurboIMAGE allows the database administrator to create a directory called a database-access file. This access file defines the location of the database and the authorized users. Only the administrator need be concerned with actual locations; TurboIMAGE and NS/3000 or DS/3000 handle the rest. This flexibility allows the administrator to relocate databases without affecting the users' operating procedures or changing any application programs.

TurboIMAGE Specifications

- Data item names per database: 1023
- Data items per data entry: 255
- Data sets per database: 199
- Detail data sets per master set: 16
- Master data sets per detail set: 16
- Search items (keys) per detail data set: 16
- Maximum entry size: 4096 bytes
- Entries per data set: 2 billion (with an assumed blocking factor of 255)
- Entries per chain: 2 billion
- Characters per database name: 6
- Characters per password: 8
- Characters per data set name: 16
- Characters per data item name: 16

Database Compatibility

TurboIMAGE applications may be run in Native Mode on the 900 Series of HP 3000 systems or in Compatibility Mode. IMAGE applications may therefore be developed which share a common source code across the entire HP 3000 family. See the TurboIMAGE/XL data sheet for further details.

IMAGE is also available on the HP 1000 family and selected models of the HP 9000 family. Although very similar in nature, their external specifications and internal storage format is different. The reference manuals for the individual IMAGE products indicate the features and specifications supported by the particular IMAGE model. It is possible to move data from one computer to another, if the differences between the IMAGE databases are taken into consideration in the design phase.

System Environment

TurboIMAGE/V is supported on MPE V HP 3000 based systems (U-MIT or later).

Ordering Information

The right to use TurboIMAGE/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

TurboIMAGE/V is bundled with the Fundamental Operating Software (FOS) on all MPE V and the 900 Series of HP 3000 systems.

Documentation

Part Number	Document Description
32215-90050	TurboIMAGE/V Reference Manual
32215-90042	QUERY/V Reference Manual

Customer Training

A one week class on TurboIMAGE and QUERY/V is available through the local sales office. The course includes instruction and labs. Training is provided on:

- Creating a TurboIMAGE database using the Data Definition Language
- Practice with the Data Manipulation Language
- Using QUERY against a TurboIMAGE database

Contact the local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Number 36020A

TurboIMAGE DBchange/V

For MPE V Based HP 3000 Systems

Easy Restructuring for TurboIMAGE/V Databases

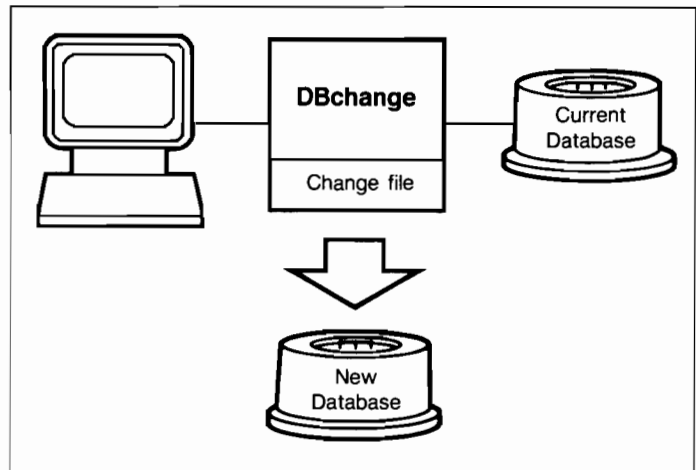
DBchange/V is the interactive utility which allows dynamic restructuring and capacity expansion of a TurboIMAGE database. Database changes are input using a menu format and executed either on-line or in batch mode.

DBchange allows easy screen access and input of database restructuring requests for TurboIMAGE databases. The central menu is shown on the next page. Because multiple restructuring changes can be requested during one DBchange session, there is no need to run separate programs for each restructure feature. DBchange allows you to review multiple change requests before actual updates are processed.

DBchange provides the database administrator with an important tool to aid in the use and support of TurboIMAGE databases. Additional benefits include ease of use, and batch capabilities. DBchange also includes the DICTDBA, DICTDBU, and DICTDBL database utilities to provide a complete set of database management capabilities.

Key DBchange Features

- Interactive forms access allows flexible and friendly database manipulation.
- Two-stage update process allows input and review of changes before actual update.
- Multiple changes can be made to the same database in one DBchange session.
- Database restructure requests are input and stored in a change file. DBchange uses the change file to process database modifications.
- Modifications can be processed immediately or scheduled for later batch processing.
- TurboIMAGE database modifications can be made in real-time, without requiring a DBUNLOAD and DBLOAD.
- On-line help facility provides information about commands and parameter requirements.
- The Root File is checked for path information inconsistencies, allowing DBchange the option to make corrections.



Broad Restructure Functionality

Interactive DBchange commands allow the user to:

- Rename a database
- Copy a database
- Change database security
- Restructure a database
- Print schema
- Change set capacity and blocking factors

All TurboIMAGE and MPE file security and capacities are respected. DBchange verifies that the user has proper access before it allows changes.

Additional Utilities

Included as part of the DBchange package, are three utilities to facilitate the control, maintenance, and restructuring of TurboIMAGE databases. These are the DICTDBA, DICTDBU, and DICTDBL utilities.

DICTDBA

DICTDBA is a database utility which checks the integrity of synonym chains and identifies any broken chains. It generates reports on synonyms and chains for master sets and on chain statistics for detail sets.

```

DBCHANGE Main
Data base name:

f1 - Rename data base      new data base name:
f2 - Grant data base copy access to      user:
                                           group:
                                           account:
f3 - Security Screen
f4 - Restructure Screen
f5 - Print schema
f6 - Perform restructure

Rename Grant Security Restruct
Database Copy Screen Screen
Print Perform Help Exit
Schema Restruct

```

DICTDBU

DICTDBU is a database utility for unloading the contents of an existing TurboIMAGE database. With DICTDBU the database can be unloaded "chained" to either a disc or tape file, thereby organizing the records for more efficient chained access. DICTDBU also provides the capability to only unload selected data sets.

DICTDBL

DICTDBL is a database utility for loading the stored contents of a TurboIMAGE database. This utility simplifies and speeds up the database load function when only a few sets must be loaded.

System Environment

DBchange is supported on MPE V based HP 3000 systems (UB delta 2-MIT or later) and in Compatibility Mode on the 900 Series of HP 3000 systems.

DBchange also requires TurboIMAGE/V database management system and VPLUS/V screen manager.

Ordering Information

The right to use DBchange/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
36020A	DBchange/V Right-to-Use product
36020R	DBchange/V Right-to-Copy product with sublicense

Documentation

Part Number	Documentation Description
36020-91001	DBchange/V User's Guide
32244-90012	Database Utilities Reference Manual

HEWLETT-PACKARD

Product Number 36914A

TurboIMAGE Profiler/V

For MPE V Based HP 3000 Systems

The Performance Tuning Tool for TurboIMAGE/V

TurboIMAGE Profiler is a diagnostic analysis tool which works with TurboIMAGE to guide users in tuning databases and application programs.

Profiler provides performance and usage statistics which can be used to help determine the optimal design of a TurboIMAGE database. Profiler uses trace data gathered while TurboIMAGE applications are running; and via a friendly, forms-driven interface, provides the capability to request a wide range of statistical reports. With Profiler, programmers and the database administrator now have a powerful tool for finding ways to improve the performance of TurboIMAGE based applications.

Key Profiler Features

- Collects trace data at either job/session or system wide level.
- Collects trace data only during user specified time intervals.
- Consolidates trace data into manageable statistics.
- Report format flexibility provides capability to only view the most relevant statistics.
- Provides default or custom tailored reports.
- Provides interactive forms interface.
- Runs without operator intervention.

Report Categories

Profiler provides a variety of statistics to help the user determine the best TurboIMAGE database design. Default or customized reports can be generated in each of these categories.

Intrinsic Performance Statistics

These reports show the amount of time and system resources that individual TurboIMAGE intrinsic calls are using. This insight facilitates making performance trade-offs in designing and fine tuning applications.

PROFILER/3000 STATISTICS SELECTION STAT

Trace interval name _____

Output file name TDAOUT

Select statistics type

- I - Intrinsic activity
- T - Transaction activity
- B - Buffer activity
- P - Path activity
- L - Lock conflict activity
- A - Run all reports with default

COMMAND _____

LIST DIR REFRESH GO TO MAIN PREV FORM HELP EXIT

Statistics Selection Screen

Transaction Performance Statistics

These reports provide statistics on the performance characteristics of transactions. A transaction is a sequence of intrinsic calls defined through the use of DBBEGIN and DBEND.

Buffer Effectiveness Statistics

These statistics show how buffers are being utilized by TurboIMAGE. This information helps in selecting the optimal buffer specifications for best performance.

Path Effectiveness Statistics

Statistics on the usage of paths are provided in these reports. These reports help to identify paths whose chain maintenance overhead overshadows their usefulness. This information also facilitates the correct selection of primary paths.

Lock Conflict Analysis

These reports provide information on lock conflicts and their effect on an application's performance. These reports also identify transactions which are impacting performance by holding locks for too long.

For existing application programs which are experiencing performance problems, Profiler can help identify bottlenecks and other problem areas. Profiler can also serve as a valuable debugging tool by identifying which intrinsic calls are being executed in a test run. Knowing which intrinsic calls are being executed helps the application designer identify which sections of the code were tested.

System Environment

TurboIMAGE Profiler/V is supported on MPE V based HP 3000 systems (U-MIT or later).

Profiler/V also requires TurboIMAGE/V and VPLUS/V.

Ordering Information

The right to use Profiler on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product

Number	Product Description
36914A	TurboIMAGE Profiler/V Right-to-Use product
36914R	TurboIMAGE Profiler/V Right-to-Copy product with sublicense

Documentation

Part

Number	Document Description
36914-91001	TurboIMAGE Profiler/V User's Guide

HEWLETT-PACKARD

Product Number 32208A

KSAM/V

For MPE V Based HP 3000 Systems

Keyed File Management System

KSAM (Keyed Sequential Access Method) allows you to create and maintain disc files whose records are accessed by the value of the key fields within the data records. Each data record contains one primary key field and may include up to 15 alternate key fields. Data records are written to a KSAM file in any order without regard to a key sequence, although they may be presorted if desired. Records are accessed sequentially or randomly by primary or alternate key value, by logical record number, or in chronological (physically sequential) order. Duplicate key values are allowed, and records can be accessed by generic keys (partial key values) or by approximate keys.

Key KSAM Features

- Multiple keys: one primary and up to 15 alternate keys
- Duplicate key values allowed
- Retrieval by generic key value or by approximate match
- Access from COBOL, RPG, FORTRAN, Pascal, BASIC or SPL
- Fixed or variable length data records

KSAM Components

KSAM consists of the following two logical components:

KSAM procedures and intrinsics. To assist in accessing KSAM data records, special sets of COBOL procedures, BASIC procedures, and MPE file system intrinsics (for FORTRAN, Pascal, and SPL) have been provided.

Note: Most of the standard MPE file system intrinsics can be used with little or no parameter changes to access and manipulate KSAM files.

KSAM utility program (KSAMUTIL). This utility program provides a set of commands designed specifically for creating and manipulating KSAM files.

BUILD	Creates a KSAM file consisting of a data file and a key file.
ERASE	Clears the contents of a KSAM data file and resets the pointers in the associated key file.
PURGE	Deletes a KSAM file from the system.
RENAME	Changes the name of a KSAM data file or key file.

SAVE	Saves a session/job temporary KSAM file as a permanent file.
VERIFY	Displays the current status of a KSAM file.
HELP	Displays a description of all the KSAMUTIL commands.
EXIT	Terminates KSAMUTIL and passes control back to the MPE command interpreter.

KSAM File Structure

A KSAM file consists of two physical disc files: a key file and a data file. A KSAM data file contains all the data records. The associated key file contains one or more sets of entries that maintain the primary and alternate logical sequences of the data records. When a data record is added to the data file, a key entry is added to the associated key field in the new data record. These key entries are dynamically added in ascending order to maintain the sequential nature of the key file. Diagram A illustrates the functional relation between a KSAM key file and data file.

Accessing KSAM Files

Data records in a KSAM file can be retrieved, updated, or deleted sequentially or in random order by key value through the key file. They can also be accessed as they were written, in chronological order, or by record number without using the key file. KSAM files can be accessed:

RPG/V	Through file specifications and chaining operations.
COBOL II/V	Through the Indexed I/O Module defined within the COBOL 74 ANSI Standard.
FORTRAN/V	Through call statements to MPE file system intrinsics or to the special set of KSAM procedures.
Pascal/V	Through MPE file system intrinsic calls.
BASIC/V	Through call statements to a special set of KSAM procedures designed specifically for use with KSAM files.
SPL/V	Through MPE file system intrinsic calls.

KSAM files can be opened for shared access by more than one user, or access to a KSAM file can be restricted to a single user.

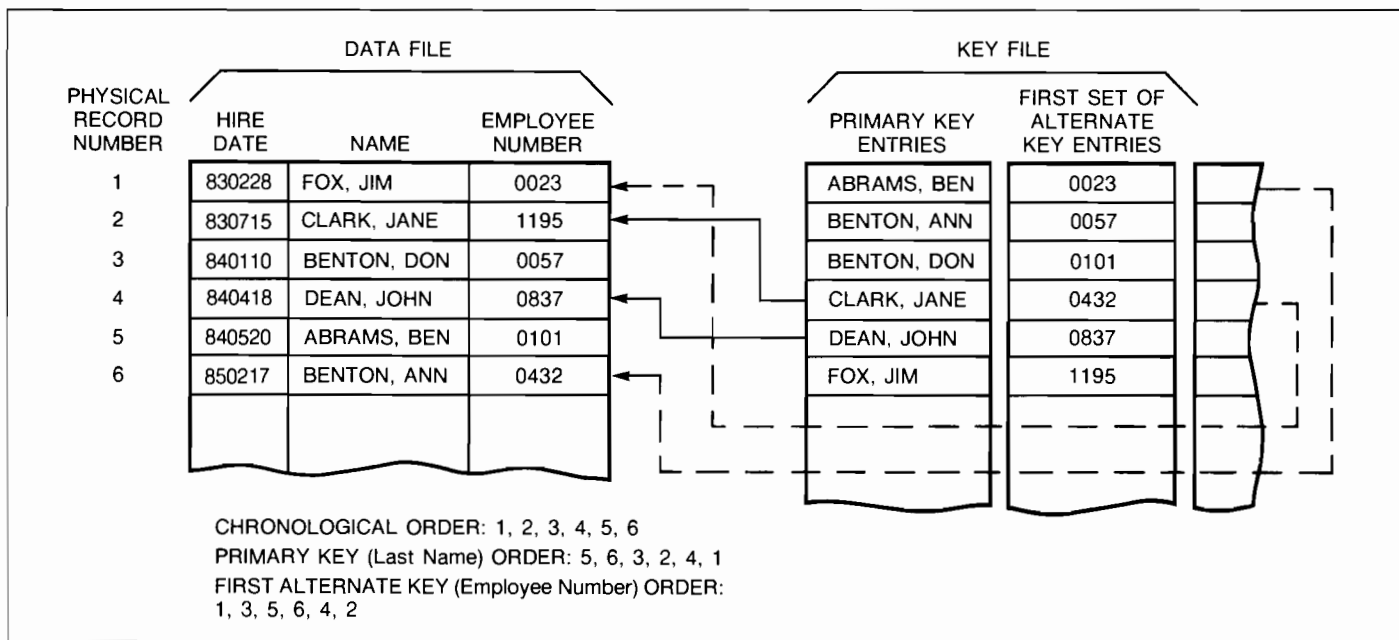


Diagram A

Key File Structuring

KSAM automatically performs any key file restructuring required to accommodate newly added records to the associated data file. This restructuring is invisible to the programmer.

Data Types

Data fields, including key fields, may contain the following types of data: integer, double integer, real, four-word long real, byte string, packed decimal, and numerical display.

Security

All MPE file system security provisions also apply to KSAM files.

KSAM File Recovery

A KSAM file that has been opened for read-only access during a system failure can be reopened after the system is brought back up, and does not need to be recovered.

If the system fails when a KSAM file is open for any other type of access except read-only, the file can only be reopened after it is recovered. The file is easily recovered in most cases by running the KEYINFO command of the KSAMUTIL program.

System Environment

KSAM/V is supported on MPE V HP 3000 based systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

The right to use KSAM/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

KSAM/V is included in the Fundamental Operating Software on all HP 3000 Computer Systems.

Documentation

Part Number	Documentation Description
30000-90079	KSAM/V Reference Manual

HEWLETT-PACKARD

Product Number 30302A

Silhouette/V

For HP 3000 Computer Systems

Silhouette/V is a software product that automatically duplicates IMAGE and TurboIMAGE databases on multiple HP 3000's over network links. Silhouette requires no modification to HP 3000 application software and is transparent to application users. Silhouette provides users with the flexibility to shadow the databases from one primary system to multiple secondary systems, from multiple primary systems to one secondary system, and for two systems to mirror the databases of each other. The product is easy to use and customer installable.

Benefits

High Availability. Silhouette/V maintains up-to-date duplicate IMAGE or TurboIMAGE databases on a secondary HP 3000. If a hardware failure occurs on the primary system, users can switch to the secondary system and continue operations. Data integrity is maintained because Silhouette automatically removes and reports all incomplete transactions to each user. These transactions may involve updating either single or multiple databases.

Application Availability During Backup. Because duplicated IMAGE or TurboIMAGE databases exist on the secondary system, customers may do backup at their convenience on the secondary system. This means that users have access to their production applications 24 hours a day on their primary system.

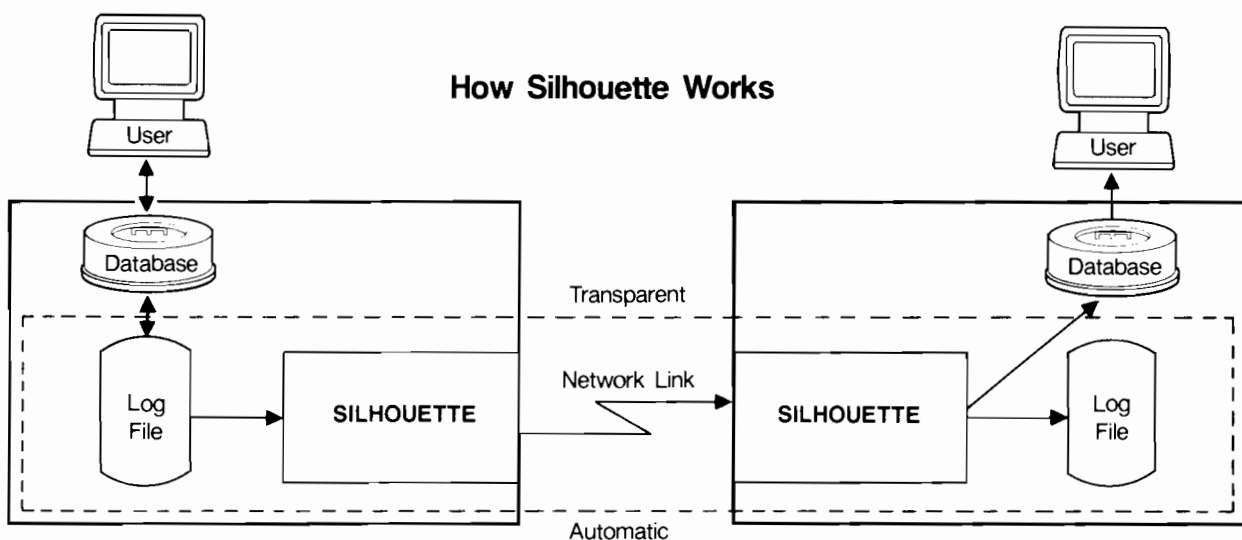
Load Balancing Across Systems. Silhouette allows inquiry-only access to the duplicated IMAGE or TurboIMAGE databases on the secondary system. Thus, customers may move their read-only users from the primary system to the secondary system to reduce primary system workload.

Functional Description

Silhouette/V duplicates IMAGE or TurboIMAGE databases by transporting the log file of the primary system over network links to the secondary system. Silhouette then automatically updates IMAGE or TurboIMAGE databases of the secondary system. To use Silhouette, the operator simply specifies the databases to be mirrored, enables database logging on the primary system, specifies the time frequency that Silhouette examines the log file, specifies how far Silhouette may get behind in log file transmission before Silhouette requests the operator to mount a magnetic tape for storing records and specifies which network link(s) will be used by Silhouette.

Primary System Failure. Silhouette notifies the primary system's operator when the primary system fails. The operator then can issue a SWITCH command to transfer primary system status to the secondary system.

How Silhouette Works



Secondary System Failure. Silhouette notifies the operator that the secondary system is not receiving records and begins logging the untransported records to a buffer, the size of which is set by the operator. If the number of records exceeds this "high water mark", Silhouette begins to log the records to magnetic tape. The operator then uses either the buffer or magnetic tape to update the secondary system's mirrored IMAGE or TurboIMAGE databases.

Network Link Failure. If a system has more than one network link and one fails, Silhouette automatically spreads out record transmission over the remaining links. If the last link fails, Silhouette notifies the operator and begins logging records to the buffer. Again, if the number of records exceeds the "high water mark", Silhouette writes the records to magnetic tape. The customer may use either the buffer or magnetic tape to update the shadowed databases.

In all cases where a failure occurs, Silhouette automatically removes any incomplete transactions to single or multiple databases and creates reports which list both the incomplete transactions and the last completed transaction so that users may make certain that the duplicated database prior to the failure matches the mirrored database after the failure.

Performance

Silhouette/V overhead has minimal impact on the primary system and moderate impact on the secondary system. The following factors may affect secondary system performance:

- Size of the secondary SPU
- Secondary system processing requirements not related to Silhouette
- Number of inquiry-only users on the secondary system
- Frequency of logfile examination
- Number of network links dedicated to Silhouette

System Environment

An HP Systems Engineer will determine the most appropriate system configuration for a particular application by running the Silhouette/V Capacity Planning Tool.

Silhouette runs on T-Delta-3 and later releases. Silhouette requires two or more SPUs, each of which must support the above-mentioned operating system, and at least one data communications link and service which will be dedicated to Silhouette. Silhouette will operate with any IMAGE-based application software and any peripherals which are supported by the SPU and operating system.

Ordering Information

The right to use Silhouette/V on an HP 3000 system is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Silhouette/V support falls under the Data Management Support Category. Silhouette support is in addition to the Data Management category of support products.

Product Number	Product Description
30302A	Silhouette/V Right-to-Use product
30302M	Silhouette/V Right-to-Copy product without sublicense
30302A+S00	Silhouette/V SMS Support
30302A+W00	Silhouette/V SMS-Extended Support
99083B+C00	Data Management AMS Support
99083B+V00	Data Management Extended Support
99083B+Q00	Data Management MUS Support

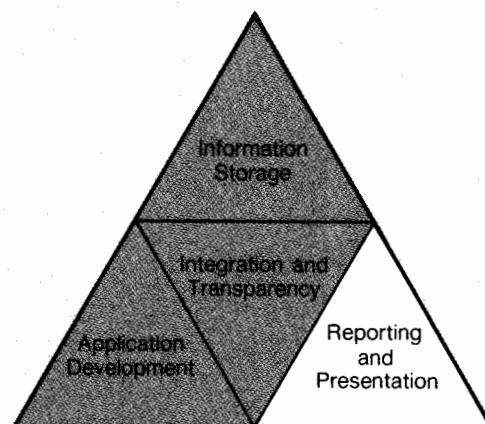
Documentation

The Silhouette/V Reference Manual (P/N 30302-90001) provides answers to questions concerning installation, normal use, failure recovery, command syntax and error messages in an easy to understand format. The manual is shipped with each copy of the product.

Customer Training

Silhouette/V is customer installable. To ensure successful implementation and operation of Silhouette, Hewlett-Packard requires either One Day SE Assistance (35032A) or One Day Specialist Assistance (35034A) for the first copy of Silhouette installed by the customer. This day of consulting helps the customer to understand the product, how to best use it for particular applications, when and how to switch operations and users to the secondary system, and provides suggestions for ensuring a successful installation.

Reporting and Presentation



Business Report Writer/V

For MPE V Based HP 3000 Systems

Report Writing Made Easy

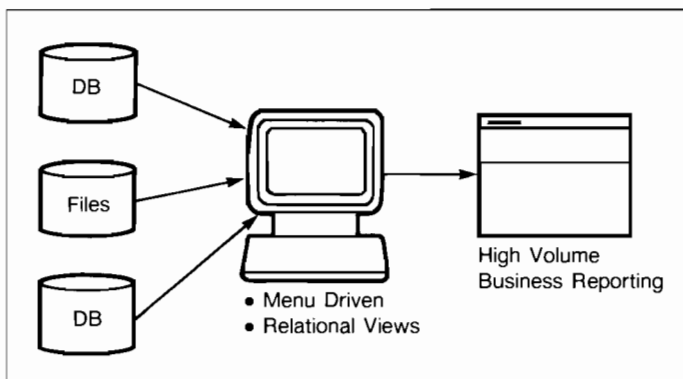
Business Report Writer/V is a high-performance report writing system for data processing professionals (programmers, system analysts, and database administrators). Comprehensive report development no longer requires numerous lines of programming code nor complex report syntax. With Business Report Writer, your EDP department can refer report development tasks to the database administrators and systems administrators. Ease-of-use separates Business Report Writer from other reporting tools. A menu-driven interface, assisted by an on-line help facility, aids users in specifying reports. The backlog of report requests is reduced since turnaround times are shorter. "Sample Layout" features allow report developers to quickly verify the format and design of reports without compiling. A help facility built into Business Report Writer guides users through logical report development and maintenance.

Reports have to do more than list data. They have to transform raw data into information. And, they have to do it quickly. Business Report Writer reduces the time required to design reports so users get the information they need faster. Good business reporting also provides the analysis to facilitate decision making. The more analysis a report provides, the sooner business people can finalize decisions. Business Report Writer's advanced features perform the analysis. Detailed calculations, date and time functions and exception reports are just a few examples.

Business Report Writer uses System Dictionary, Dictionary/V or the HP Application Dictionary. These products store data definitions, maintain documentation and define access paths to data. Business Report Writer can access TurboIMAGE/XL databases as well as IMAGE/V and TurboIMAGE/V databases, MPE files, and KSAM files. Once data is selected, Business Report Writer optimizes the access to the files and data sets using a predefined set of algorithms, ensuring maximum run-time performance.

In addition to database security and MPE file security, Business Report Writer prevents unauthorized access to report specifications, as well as report execution through its internal security system. The combination of these protective measures ensures the safety of information throughout the data processing environment.

Minimizing the programming effort required for report development results in productivity increases for the entire data processing department. Reports are more timely. Computer resources are conserved. Information and analysis, not just data, is presented for immediate decisions. And, programmers do what they do best — program development.



Key Business Report Writer Features

- "Fill in the box — see what you get" report specifications that allow immediate design verification and reduce maintenance time.
- Relational views of data that off-load the problem of "how" to get data and allow the designer to focus on the user's request.
- Multi-pass reporting (reporting on the results of other reports) allows the output file of a report to be accessed by other reports or applications.
- Powerful calculation features, for specifying equations and business functions, provide the analysis that decision makers need from a report writing system.
- Flexible report formatting presents information and analysis that adds value to the decision process.
- Integration with System Dictionary, Dictionary/V and HP Application Dictionary for reporting data from several applications.
- Performance tuning that allows users to optimize performance and conserve system resources.
- Remote access for selecting data from a remote HP 3000.
- On-line help facility that guides users through report design and maintenance.
- Run-time programs as part of the MPE V Fundamental Operating System.
- Intrinsics for programmatic access to reports, with no need to exit an application.
- Integration with the VIRTUOSO COBOL Code Generator.
- Conversion utility to convert Report/V or Inform/V report definitions to Business Report Writer Programs.

System Environment	Customer Training
---------------------------	--------------------------

Business Report Writer/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Business Report Writer also requires VPLUS/V (and VPLUS/V compatible terminals) and System Dictionary, Dictionary/V or the HP Application Dictionary.

Ordering Information

The right to use Business Report Writer/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
36070A	Business Report Writer Right-to-Use product
36070R	Business Report Writer Right-to-Copy product

Documentation

Part Number	Document Description
36070-60002	Reference Manual Kit
5954-6217	Product Evaluation Guide

A four day class for Business Report Writer is available through the local sales office. Each course includes instruction and labs.

Course Number	Course Name
36070AB	Business Report Writer User's Course



HEWLETT-PACKARD

Product Number 32245A

Report/V

For MPE V Based HP 3000 Systems

Report Writing Language

Report/V is a nonprocedural, report writing language for programmers. The product includes a high-level language that creates reports, a compiler, and a processor to execute the reports. High-level statements produce report formats, perform calculations, define selection criteria, and access the data. For more functionality and a menu-driven interface a follow-on product to Report/V is available. This product is Business Report Writer, the reporting standard for MPE V and MPE XL computer systems. Within Business Report Writer/V is a utility that automatically converts Report/V statements into Business Report Writer menus. For more information see product 36070A.

Report/V is linked to the standard dictionaries, databases and files found on the MPE V systems. Report, together with Dictionary/V, resolves data definitions, structures, and access paths transparently. Access to TurboIMAGE/V databases, KSAM/V, and MPE V files puts programmers within reach of all data resources quickly and easily. It is also possible to access TurboIMAGE/XL databases, KSAM/V and MPE XL files when running on 900 series HP 3000 computers.

The data is protected. Security can be defined to prevent unauthorized access to reports files, report formatting statements, and databases. Access modes can be set to ensure data integrity.

Good business reporting involves more than presenting the facts. Report has the power to perform advanced calculations, report statistical data and extract information based on logical conditions (IF-THEN-ELSE). These capabilities, combined with Report's formatting features, deliver the information and analysis that decision-makers need.

The process of report creation can be a real bottleneck in a rigid development cycle. Yet, with Report, programmers no longer have to write large, complicated programs for producing reports. Instead, report development times are reduced using the high-level formatting statements of Report. With these productivity gains, development time is shorter than those of other programming languages.

Key Report Features

- High-level, free format language provides comprehensive report development without tedious programming effort.
- Access to TurboIMAGE/V and TurboIMAGE/XL databases, MPE V, MPE XL, and KSAM/V files support a common, transparent path to various data resources.

- Interface with DICTIONARY/V transparently resolves data definitions, structures, and access paths reducing report development time.
- Advanced reporting features that deliver an analysis of the information to facilitate decision making.
- Security protection for report definitions and execution that prevents unauthorized access to sensitive information.
- Definable modes for opening TurboIMAGE databases.
- Conversion utility to convert Inform/V report definitions to Report/V programs.

Fast, Simple Report Creation

With Report, a simple report, such as a list of production information sorted by product number, needs only the following code:

```
report PROD;  
report title "PRODUCT LIST";  
detail PROD-NUM: PROD-NAME:  
        PROD-PRICE;  
sort PROD-NUM;
```

This code produces the following output:

Product	Product Description	Price
10	RACER, 27", 10 SPEED, BLUE	\$155
20	RACER, 27", 10 SPEED, BLACK	155
30	RACER, 27", 10 SPEED, RED	155
40	RACER, 27", 10 SPEED, GREEN	155
50	SPEEDSTER, 27", 5 SPEED, BLUE	150
60	SPEEDSTER, 27", 5 SPEED, BLACK	150
70	SPEEDSTER, 27", 5 SPEED, RED	150
80	SPEEDSTER, 27", 5 SPEED, GREEN	150
90	RACER, 23", 10 SPEED, BLUE	145
100	RACER, 23", 10 SPEED, BLACK	145
110	RACER, 23", 10 SPEED, RED	145
120	RACER, 23", 10 SPEED, GREEN	145
130	SPEEDSTER, 23", 5 SPEED, BLUE	140
140	SPEEDSTER, 23", 5 SPEED, BLACK	140
150	SPEEDSTER, 23", 5 SPEED, RED	140
160	SPEEDSTER, 23", 5 SPEED, GREEN	140

Report retrieves the headings, edit masks, data definitions, and access paths from Dictionary/V to create the default layout seen above. Programmers need not know where any of the information is stored.

Advanced Features

Decision makers don't like reports cluttered with detail information. Yet, providing a summary frequently requires a detailed analysis. With the addition of a few more statements, Report programmers can create more complex, customized reports.

Extracted items can be used in calculations and the results included in the report. Arithmetic operations can be performed on data using logical conditions. Specific selection conditions can be defined within the Report statements or selection parameters can be passed to the user at run time. IF-THEN-ELSE constructs in Report allow programmers to build logical conditions into reports which provide the concise information users request.

For example, a sales manager in a small company anticipates a strong order month. Yet a key business issue exists: How much additional cash will be required to pay the sales force their bonus commissions (for those over 100% of quota)? Using Report, programmers can specify a report that requests sales volumes for those sales reps . . . IF they have exceeded quota for the month . . . THEN calculate the amount of excess commissions to be paid to each employee . . . AND summarize the total commissions paid. In a small company, this type of analysis might be used by sales management to inform accounting of additional cash requirements needed to pay the bonus commissions.

EXAMPLE OF CONDITIONAL STATEMENT:

```
IF(ORDERS) > (QUOTA) THEN LET (BONUS) = .10 * (ORDERS) - (QUOTA)
```

Data from this report can be compared against other data items. The sales manager could compare this month's order performance with previous periods, for trend analysis.

Report also provides a wide range of options to facilitate report presentation. Data items can be sorted in ascending or descending order. Data can be logically grouped and titles and headings can be specified for each group. Summaries are automatically included for each grouping as well as the entire report. Additional features include: Calculation or averages; Minimum and maximum values, and statistical information such as ranges.

The following statements show an example of Report's advanced features that extract a specific set of data and summarize them into an easy-to-read report.

```
report BONUS;
report title "BONUS COMMISSION REPORT", center;
group(1) title "PRODUCT-LINE", LINE=1:
    PROD-NO, nohead:
    PROD-NAME, LINE=1;
define (BONUS) I(6), INIT=0;
detail SALESREP: ORDERS: QUOTA: BONUS;
group(1) summary "TOTAL":
    total (ORDERS), align:
    total (QUOTA), align:
    total (BONUS), align:
sort(1) PROD-NO;
block
    if (ORDERS) > (QUOTA)
    then let (BONUS) = .10 * (ORDERS) - (QUOTA)
    else let (BONUS) = 0;
blockend;
select PROD-NO;
```

Below is an example of the report produced from the statements. The "select" statement used above allowed the user to specify a specific product line for the report.

BONUS COMMISSION REPORT			
PRODUCT LINE	45	LOCAL AREA	NETWORK INTERFACES
SALESREP	ORDERS	QUOTA	BONUS
ALEXANDER	\$ 45,000	\$ 39,000	\$ 600
JOHNSON	\$ 62,000	\$ 60,000	\$ 200
HILTON	\$ 38,400	\$ 38,000	\$ 40
DOLEY	\$ 26,000	\$ 20,000	\$ 600
TOTAL	\$171,400	\$157,000	\$1,440

Upgrade Path to Business Report Writer

Customers can convert reports that have been written in Report/V by using the conversion utility supplied within Business Report Writer. Any features that do not exist in Business Report Writer and cannot be translated will cause a warning message to be printed. After conversion, reports can be further enhanced using Business Report Writer's expanded feature set. Report/V customers that purchase Business Report Writer will also receive upgrade credits.

System Environment

Report/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product

Number	Product Description
32245A	Report/V Right-To-Use product
32245R	Report/V Right-To-Copy product

Documentation

The right to use Report/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase agreement.

Document

Number	Document Description
32245-90001	Report/V User's Guide

Customer Training

A training class for Report/V is available through the local sales office. Each class includes instruction and labs.

Course

Number	Course Name
35013C	Dictionary/Inform/Report Customer Training

HEWLETT-PACKARD

Product Number 32246A

Inform/V

For MPE V Based HP 3000 Systems

An End User Facility

Inform is an interactive inquiry and report generation facility oriented towards end-users (for example: managers, purchasers, clerks, or secretaries). Inform provides an interactive menu facility that allows access to data and the generation of reports in a fraction of the time needed by traditional methods. Inform and Dictionary/V work together to offer a high degree of data access transparency. Inform accesses the dictionary for definitions and locations of data stored in TurboIMAGE/V databases, KSAM/V, and MPE V files. It is also possible to access TurboIMAGE/XL databases, KSAM/V and MPE XL files when running on 900 Series HP 3000 computers. End users can access information without knowing how or where it is stored.

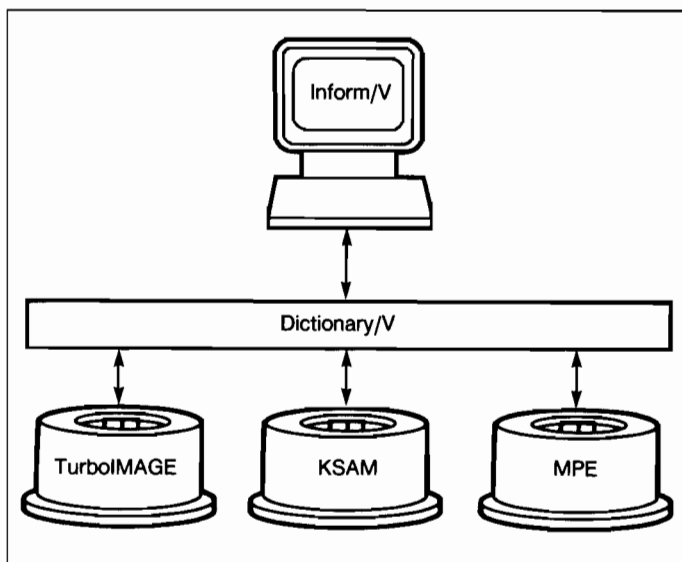
Providing quick access to on-line information increases end-user productivity. By eliminating most of the programming effort previously required to develop reports, the programming staff's productivity is increased as well. Programmers can spend more time on developing and maintaining application systems and less time on report writing.

Inform protects the integrity of corporate data by allowing read-only access to databases and files. In addition, security can be assigned to individuals preventing unauthorized access to data that is classified as sensitive.

Inform can generate reports either on-line at the terminal, or in batch mode. In batch mode, Inform can perform all of its work in the background and return terminal control back to the end user immediately.

Key Inform Features

- Fast, friendly, direct interactive link between end users and data stored in HP 3000 databases and files
- Inform security limits user access to data only if authorized
- Designed specifically for the end user, off-loading the programmer from the time consuming task of report writing
- Immediate report production provides information as it is needed
- Report definitions can be saved and reused, which prevents monotonous report redefinition
- Access to multiple TurboIMAGE/V and TurboIMAGE/XL databases, KSAM/V, MPE V, and MPE XL files means reports can be created from more than one file source.



- Automated report formatting supports easy report generation
- User-definable report formatting supports tailor-made reports
- Friendly menu interface
- Automated resolution of data and file definitions through Dictionary/V provides data access transparency

Functional Description

Inform is an interactive inquiry and report generation facility designed specifically for end users. Access to data is through logically related groups, called Inform Groups, which are defined in Dictionary/V by the dictionary administrator. Using groups, Inform has the capability to access multiple TurboIMAGE/V databases, KSAM/V, and MPE V files. Through the Inform groups, end users are provided with their personal view of the data — no matter where the data resides. TurboIMAGE/XL databases and MPE XL files can also be accessed when running on the 900 Series computers. Inform also provides access to TurboIMAGE datasets. The reporting and inquiry functionality that is available for Inform/V groups is also available for TurboIMAGE datasets.

Below is an example of the Data Organization Menu:

```
INFORM/V                      HP32246A.04.00 - (C) Hewlett-Packard Co. 1987

DATA ORGANIZATION

INFORM/V GROUPS:

1: DISPLAY GROUP NAMES
2: SELECT A GROUP

DATA BASE ONLY:

3: DISPLAY DATA BASE NAMES
4: SELECT A DATA BASE

TYPE A NUMBER>
```

In this menu, the end user selects whether the report is to be generated from an Inform Group or a single database.

In addition to database and MPE file security, Inform offers security through the Inform Groups. Access can be granted on an entire Inform Group of data or it can be restricted to a subset of fields within the Inform Group. For example, a manager can be granted access to the entire Inform Group where a clerk may have access only to certain fields within the Inform Group.

In addition to choosing what information is to be contained in the report, end users can specify how it will be presented. Some of the formatting capabilities include:

- Whether to sort and subdivide data in ascending or descending order
- What data is to be used as a selection criteria
- What to include in group or report summaries: for example totals, averages, counts, maximums, and minimums
- New data to create based on other items in the Inform Group

Inform prompts for specific information to include in the report. It is possible to select information based on a range or pattern of values, Inform can find the information based on just a few characters. The screen in the next column is an example of the report specification menu. Displayed at the top of the menu are the fields that have been defined in the Inform Group named PRODUCTS. An end user would use this menu to create a report that utilized either one or more of the fields displayed.

Once the report has been defined, it is possible to store the definitions for future use in a report catalog. These stored reports can be used to regenerate the report or their definitions can be modified whenever the reporting needs change.

```
INFORM/V                      HP32246A.04.00 - (C) Hewlett-Packard Co. 1987

DATA NAMES IN PRODUCTS

1: PRODUCT NAME      3: PRODUCTION COST      5: PRICE
2: PRODUCT NUMBER    4: DESCRIPTION

TYPE NUMBERS(S) FOR DATA NAME(S):

TO INCLUDE IN REPORT> 2,1,5
TO SORT BY> 2
TO SUBDIVIDE BY>
TO SUMMARIZE FOR:
FOR REPORT SUMMARY>
FOR SELECTION CRITERIA> 2

REPORT TITLE> PRODUCT LIST

PREPARING REPORT

PRODUCE THE REPORT(Y/N)?> Y
```

End users can create new data items by adding, subtracting, multiplying, and dividing other data items and constants. Edit masks can also be specified for better presentation of data items.

For quick reporting, Inform provides default report formats. Or, it is possible to change titles or headings, to duplicate lines or add blank lines, as well as set the page width and length of the terminal and the printer. Inform also lets the end user preview the report layout before the report is generated.

Inform/V and Report/V, Business Report Writer/V

Inform/V report definitions can be converted to Report/V or Business Report Writer/V programs and further enhanced. Conversion utilities within Report/V and Business Report Writer/V provide this functionality. This allows end users to use Inform to layout simple reports. If more features are desired Inform reports can be converted and further developed using Report/V or Business Report Writer/V.

System Environment

Inform/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Inform/V also requires TurboIMAGE/V, or TurboIMAGE/XL, VPLUS/V, SORT/V, Dictionary/V, and system support of Native Language Intrinsics.

Ordering Information

The right to use Inform/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
32246A	Inform/V Right-to-Use product
32246R	Inform/V Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
32246-90001	Inform/ User's Guide

Customer Training

Hewlett-Packard offers both self-paced training and classroom instruction for Inform/V. Each course includes instruction and labs.

Course Number	Course Name
35013C	Dictionary/Inform/Report Customer Training
32246-90003	Inform/V Self-Paced Training Course

The on-line help facility provided with Inform/V along with the Inform menu screens serve as an additional guide to learning the product.

Contact the local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Number 32216A



QUERY/V

For MPE V Based HP 3000 Systems

Programmer's Access Tool for TurboIMAGE/V

QUERY/V is a database support tool included with TurboIMAGE. It is designed for use by application programmers and database administrators. Through the use of QUERY, it is possible to locate, update, and report on data items within a TurboIMAGE database.

QUERY communicates with TurboIMAGE through a command-driven interface. It may be executed interactively (from a terminal) or in a batch mode such as a stream file. Output may be directed to a terminal or to any other output device (e.g., line printer, tape, disc). Frequently used QUERY procedures may be kept and stored on disc for future use.

QUERY honors the TurboIMAGE database security system. A user must supply an appropriate TurboIMAGE password before access to the TurboIMAGE database is granted. Since QUERY supports DS/3000 and NS/3000, it is possible to access databases located on remote computers.

Full reporting capabilities on TurboIMAGE databases are available through Report/V, Business Report Writer/V, and Inform/V. Report and Business Report Writer provide a programmer with a quick and easy-to-use report writer. Inform is a menu-driven report generator designed with the end user in mind. For full product descriptions of each of these products, refer to the Report, Business Report Writer, and Inform data sheets.

QUERY provides the application programmer and the database administrator with an important tool to aid in the creation and support of TurboIMAGE applications and databases.

Key QUERY Features

- On-line help facility provides information about commands, their format, and required parameters.
- Interactive or batch interrogation allows for flexible database access.
- TurboIMAGE updating through addition, deletion, and modification of data records facilitates on-the-fly database maintenance.
- Boolean-logic provides fast access to variable data values.
- ADD command supports quick data entry of test data into new database structures.
- Extensive locating, reporting, and updating commands support the testing of new database applications and structures.

- Honoring TurboIMAGE security maintains database integrity.
- Utilizes TurboIMAGE logging to maintain logical and physical integrity of the database.
- Remote Database Access allows for database inquiry across computers.

Commands

QUERY command names can be spelled out or abbreviated. The commands consist of English language key words and parameters which can span several lines on a terminal screen. Included in the command set are:

DATA-BASE =	Specifies and opens the primary database to be accessed.
MULTIDB	Specifies and opens the secondary database to be accessed.
DATA-SETS	Specifies and opens the data sets to be accessed.
FORM	Displays the structure of the database being accessed.
FIND	Locates data entries in the database according to user defined specifications.
JOIN	Enables multiple data set retrieval.
MULTIFIND	Retrieves compound data entries from data sets specified by the most recent JOIN command.
ADD	Adds data entries to the database.
DELETE	Removes data entries from the database.
REPLACE	Modifies data items within data entries.
REDO	Edits and executes a command in the command history buffer.

>ADD PRODUCT

STOCK # = >> 12345678

DESCRIPTION = >> "ANTISMOGDEV"

Example of the ADD command

Database Administrator Tool

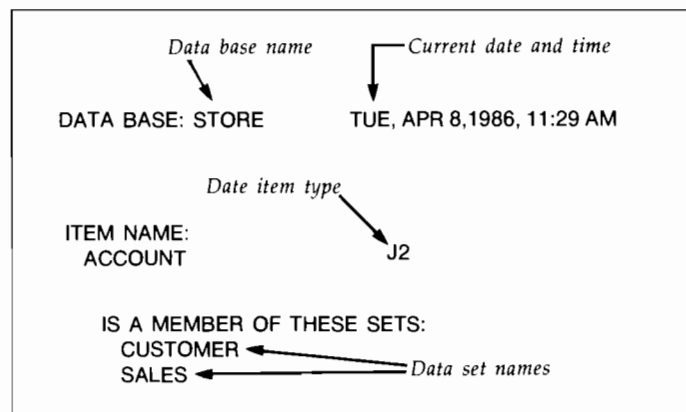
QUERY is a tool that aids the database administrator in the day-to-day maintenance of the TurboIMAGE database. The FORM command provides various pictures of the database components including:

- The database structure (excluding passwords)
- The structure of a particular data set
- A list of all the items defined in the database
- All of the path relationships between the data sets
- A list of all the data sets defined in the database including the data set capacity vs. the entry count

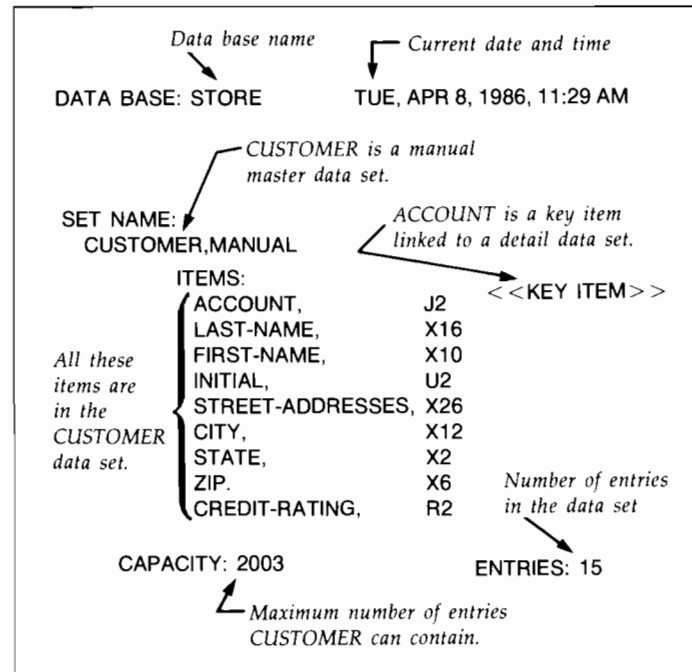
Maintenance of the database can be performed using QUERY. The ADD, DELETE, and REPLACE commands, designed for this purpose, allow insertion and deletion of data records and replacement of data item values.

Application Programmer and Database Design Tool

As a tool for application programmers and database designers, QUERY simplifies the testing process. The ADD command facilitates the creation of data in a test database. When ADD is entered, QUERY will prompt the terminal user for data item values. The user is not required to enter values for all items. With the FIND, REPORT, and UPDATE commands, QUERY will help a programmer or database designer stress test the new database. Together, these commands will allow repetitive or complex operations against the database. In addition, these commands may be grouped together to form logical database transactions which can be stored in a command file for later testing. Through the use of the MULTIFIND and JOIN commands, QUERY will support transactions which span more than one database.



FORM item example



FORM data set example

Security

QUERY adheres to all the security provisions that the database administrator includes in the TurboIMAGE database. The TurboIMAGE password, obtained from the database administrator, determines which data elements (i.e., data items and data sets) the user is allowed to access. QUERY will return an error whenever a password is entered which does not match any security level within the TurboIMAGE database. With the FORM command, it is possible to obtain a list of data elements which are accessible through a particular password.

Locating Data

QUERY is capable of retrieving all occurrences of data within a data set which meet user-specified conditions. To accomplish this, the user enters a FIND command. The FIND command string includes logical terms similar to phrases spoken in English. For example, suppose an account manager wishes to know the quantity of a particular product ordered by a specific customer. Assuming the database contains a data set called "ORDERS", the manager could locate all such orders with one FIND command as follows:

> FIND CUSTOMER-NAME IS SMITH AND
CUSTOMER-NUMBER IS 1234-5678

QUERY responds to the FIND command by locating all the data records which contain the requested data item values. In the case above, the requested data item values are SMITH and 1234-5678.

Frequently Used Procedures

Repetitive or complex operations are easily performed through QUERY's ability to execute FIND, REPORT, and UPDATE commands from a command file stored on disc. A command stored within a file is referred to as a procedure; a procedure may consist of one or more lines. QUERY provides commands for creating, deleting and listing procedures within a command file. Also, the line within a procedure may be added, deleted, or replaced.

Data Types

The following data types are converted to and from display format and error checked during QUERY I/O operations:

- one-word integer numbers
- two-word integer numbers
- two-word real numbers
- extended precision real numbers
- one-word logical values as absolute numbers
- ASCII character strings containing no lower-case
- alphanumerics general ASCII character strings
- zoned decimal numbers
- packed decimal numbers

System Environment

QUERY/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

Software

QUERY/V is included as part of the Fundamental Operating Software (FOS) on all HP 3000 computer systems.

Documentation

Part Number	Document Description
30000-90042	QUERY/V Reference Manual
32215-90050	TurboIMAGE Reference Manual

Customer Training

A one-week class on TurboIMAGE and QUERY/V is available through the local sales office. The course includes instruction and labs, with focus on:

- Creating a TurboIMAGE database using the Database Definition Language
- Practice with the Database Manipulation Language
- Use of QUERY against a TurboIMAGE database

Contact the local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Numbers 32425A, 32426A

HP Visor

For MPE V Based and 900 Series HP 3000 Systems

End User Interface to HP SQL Databases

HP Visor is an easy-to-use, terminal-based interface to HP SQL/V and ALLBASE/XL (HP SQL) databases. Visor's flexible selection of access and reporting modules enables end users or database administrators to perform queries and generate their own reports, without involving a programmer. Visor also provides powerful facilities for programmers to use in preconfiguring complex tasks, or to improve their own productivity when working with HP SQL databases.

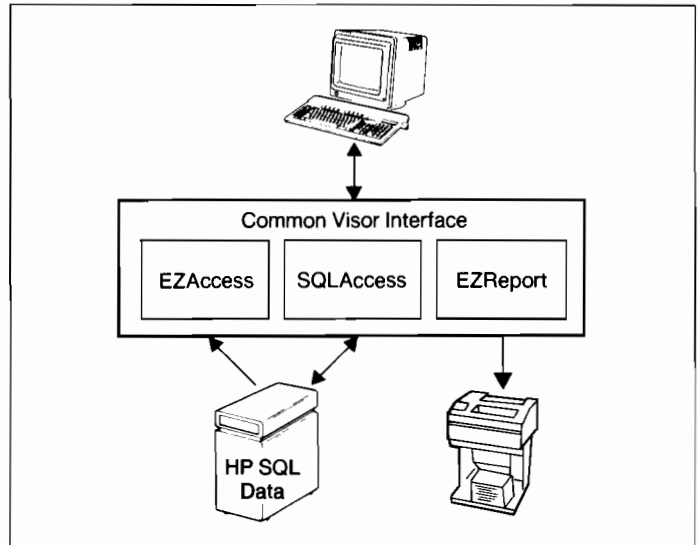
The Visor interface is forms-based, so that the user only needs to mark boxes and press function keys to step through tasks such as performing an ad hoc query or generating a customized report. Additional capabilities are provided for experienced users to directly enter Visor or SQL commands.

Overview of Modules

The three modules integrated with the Visor interface are called EZAccess, SQLAccess and EZReport. EZAccess enables novice and infrequent users of the database to easily generate queries. SQLAccess enables users familiar with HP SQL to use a screen-based editor to formulate SQL queries and perform a variety of other database operations. (SQL, Structured Query Language, is the industry standard relational database language.) EZReport formats and displays query results from EZAccess or SQLAccess, and allows users to customize report formats and calculate additional statistics using the data retrieved by the query.

Key Visor Features

- A common, forms-based interface is presented to the user; no matter what module is in use, related tasks are performed in the same manner.
- Context sensitive help, available at any time with a single keystroke, allows quick question resolution.



- Two access methods provided: menu-driven EZAccess for novice or infrequent database users; SQLAccess for users familiar with both HP SQL and the database structure.
- Queries, reports and sequences of commands can be saved for future use, thus streamlining frequently performed tasks.
- Data security is maintained: Visor relies on the HP SQL system catalog to determine each user's authority to access data.
- Default mode of operation shields novice users from complex functions and smoothly steps them through tasks that cross module boundaries.
- Experienced users can run Visor in the mode that allows them to easily select advanced features.
- Batch mode of operation is available to automate time consuming tasks.
- Data is supported in all languages with eight bit character sets: USASCII, ROMAN8 and KANA8.

EZAccess: Query Tool for Novice and Infrequent Database Users

With EZAccess, novice users and infrequent users of the database can quickly and easily learn to formulate queries and, in conjunction with EZReport, generate customized reports. No prior knowledge of either the database structure or a data manipulation language is required.

EZAccess steps users through the process of specifying the desired database (DBEnvironment), tables, columns, search values, and sort columns; the user only needs to choose among options. And, Visor's context sensitive help is available at all times with a single keystroke.

If the user would like to learn SQL, the SQLAccess module can be invoked and the current query, generated by EZAccess, will be displayed as a modifiable SQL command.

EZAccess Features

- Novice and infrequent database users are stepped through the process of creating a query; the expertise of a programmer is not required.
- Query results can be seen immediately, and queries can be iteratively refined until exactly the desired data has been obtained.
- New, calculated data columns can be defined.
- EZAccess makes "common sense" assumptions about the fields on which tables should be joined.
- EZAccess familiarizes users with SQL by generating an HP SQL query that the user can view and modify if desired.

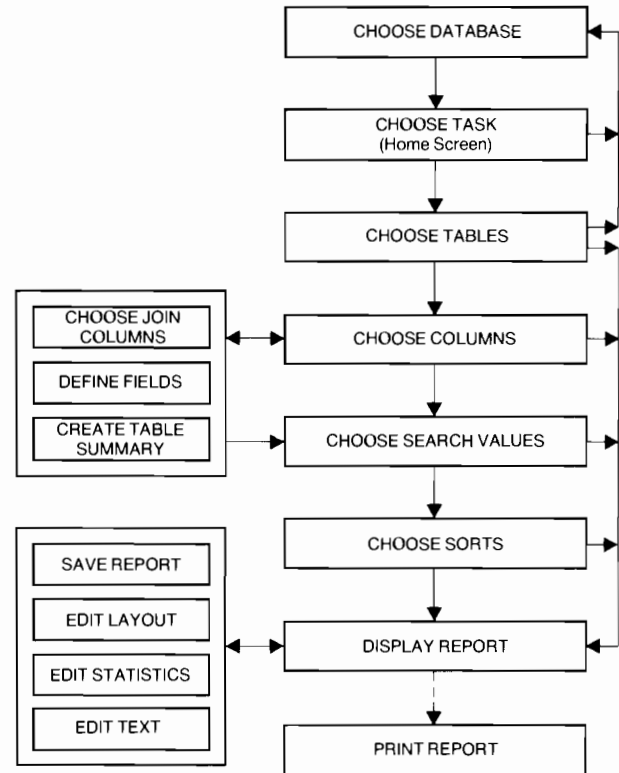
SQLAccess: Screen-Based Editor for SQL Commands

Users familiar with HP SQL can use SQL commands to retrieve and modify data, create new database objects, and maintain the database from within Visor. Commands created using the SQLAccess screen-based editor can be executed immediately, or named and saved for future use or modification.

In SQLAccess, SQL commands that perform a variety of functions beyond the capabilities of EZAccess can be created by programmers and saved for use by database end users. The commands can contain variables; Visor will prompt the user to input values when the command is executed.

SQLAccess Features

- Screen-based editor for SQL commands boosts programmer productivity with an easy way to design and test commands.
- Commands can contain variables, allowing programmers to design UPDATE, INSERT, and complex SELECT commands for end users.
- New tables and views can be created without leaving the Visor environment.



Activity flow for ad hoc query, using EZAccess and EZReport. To progress along the solid vertical path (top to bottom) the user simply marks choices in boxes on the screen and then presses the function key marked "Next Step". At any point, the "Help" function key is available to provide context sensitive help.

EZReport: The Easy Way to Create Customized Reports

Once data has been retrieved by EZAccess or SQLAccess, EZReport displays the data on the screen using a default format. From that point, the report may be customized in a variety of ways: the page layout may be changed so that columns appear in a different order or alignment; multi-line headings and footers can be added or changed; a variety of statistics (average, total, high, low, count) can be calculated and displayed at the end of the report or at specified sort breaks within the report. And, all of these choices can be accomplished with a few keystrokes, as the user only has to selectively edit the default specifications provided on full screen displays.

The reports resulting from successive modifications are displayed only on the screen until the user tells EZReport to print a hard copy or save the output to a text file. With EZReport, programmers can create and save complex report specifications for end users, and end users can easily generate their own ad hoc reports.

EZReport Features

- Database end users can design and print their own customized reports, freeing up programmer time for new applications.
- Users can check to see that the correct data has been retrieved before taking the time to customize the report design.
- EZReport provides a default report design for novice users; experienced users can easily change the layout and add headings and footers.
- Statistical functions can be calculated for any column and displayed at the end of the report or at selected sort breaks.
- The most current data is retrieved from the database each time a saved report is run.
- Reports can be printed at a system printer or local printer, or into a text file for inclusion in a document or HPDESK message.

PART SUPPLY REPORT			
PRODUCED MON, NOV 10, 1986, 10:45 AM			
PART NUMBER	VENDOR	UNIT PRICE	DELIVERY DAYS

1123-P-01	Dove Computers	450.00	30
	Eve Computers	550.00	15
	Jujitsu Microelectronics	475.00	30
	Latin Technology	500.00	20
	Seminalional Co.	525.00	15
	Space Management Systems	475.00	15

	PART NUMBER Low	450.00	
	PART NUMBER High	550.00	
	PART NUMBER Average		20
1133-P-01	Dove Computers	180.00	30
	Eve Computers	220.00	15
	Latin Technology	195.00	20
	Space Management Systems	200.00	15
	PART NUMBER Low	180.00	
	PART NUMBER High	220.00	
	PART NUMBER Average		23

Page 1			

Sample report created using EZReport

Macro-like Facility to Save Work

To enhance the productivity of all users, the Visor interface maintains four types of items: a PROFILE, SQLCOMMANDs, REPORTs and SCRIPTs.

The PROFILE allows users to customize the Visor interface to meet their individual needs; it includes specifications for the access method (EZAccess or SQLAccess), the numeric and string null-value display characters, the destination and page size for print commands, and the option of whether or not to confirm before performing tasks that alter the contents of the database.

An SQLCOMMAND is a query or other database command expressed in SQL (Structured Query Language); the command can be manually entered using the SQLAccess module or generated automatically by use of the EZAccess module.

A REPORT is the design of a report; it contains an SQLCOMMAND to retrieve the desired data, along with specifications for the page layout, additional text, and statistical calculations to be performed on the data. Each time a saved REPORT is used, the most recent data is retrieved from the database.

A SCRIPT contains a series of Visor commands to be executed in consecutive order. To minimize duplication of effort, the commands in the SCRIPT can include references to saved items, such as SQLCOMMANDs, REPORTs, and other SCRIPTs. SCRIPTs are particularly useful for automating routine functions, such as generating month-end reports and printing them at several different printers.

Visor enables the user to iteratively refine SQLCOMMANDs, REPORTs and SCRIPTs, view the results immediately, and then save the items for future use or modification. While only one PROFILE per user can be active at a time, numerous SQLCOMMANDs, SCRIPTs, and REPORTs can be named and saved for private use or to be shared with other users.

Specifications

HP Visor (maximum)

Lines per script	24
Variables per script	10
Number of saved items †	unlimited

EZAccess (maximum)

Table columns per query	15
Computed columns per query	4
Tables per join or query	3
Columns sorted per query	9

SQLAccess (maximum)

Table columns per query	15
Computed columns per query	10
Tables per join or query	10
Lines per command	24
Variables per command	10
Columns sorted per query	9

EZReport (maximum)

Width of column(chars)	240
Width of page or report	240
Lines per page ‡	999
Length of report(chars)	2.56M

† Bounded by the size of the database (DBEnvironment).

‡ Continuous output capability also provided.

System Environment

HP Visor/V runs on MPE V based HP 3000 systems (UB-delta 2-MIT or later) and requires HP SQL/V to operate.

HP Visor/XL runs on 900 Series HP 3000 systems and requires ALLBASE/XL. Visor/XL operates only on HP SQL-accessible databases. Visor/V SQLCOMMANDs, REPORTs and SCRIPTs are upward compatible with Visor/XL.

Consult your sales representative for information about the availability of Visor on HP 9000 systems.

Supported display terminals include the 2626, 2624, any of the 23xx series, and the 150 (Touchscreen Personal Computer). The Vectra Personal Computer is supported via AdvanceLink (terminal emulation mode).

Ordering Information

Software

The right to use Visor on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Product

Number	Product Description
--------	---------------------

32425A	HP Visor/V Right-to-Use product
32425R	HP Visor/V Right-to-Copy product
32426A*	HP Visor/XL Right-to-Use product
32426R*	HP Visor/XL Right-to-Copy product

* Visor/XL can not be ordered at this time.

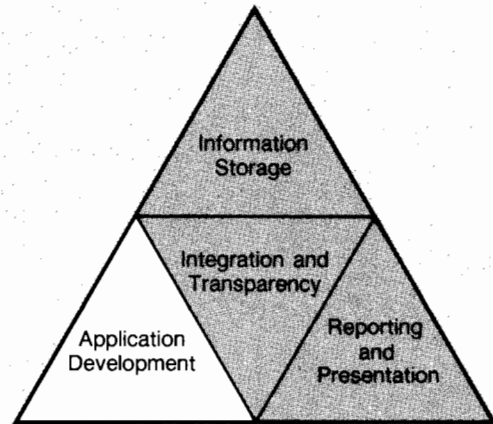
Documentation/Training

Part Number	Document Description
-------------	----------------------

32425-60001	HP Visor Tutorial
-------------	-------------------

This document primarily provides self-paced training for new Visor users; a reference information section is included. In addition, most of the information needed by users who have completed the tutorial is always at their fingertips with Visor's on-line help facility.

Application Development



VIRTUOSO Code Generator/V

For MPE V Based HP 3000 Systems

Build Your Own Application Generator

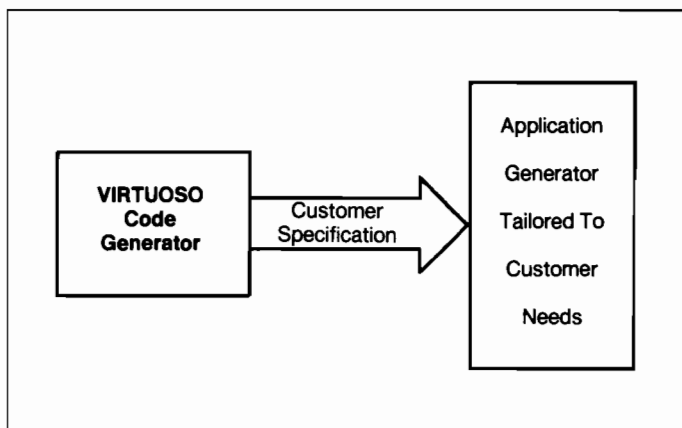
The VIRTUOSO Code Generator is a programmer's productivity tool providing easier development and maintenance of large-scale business data processing applications. The VIRTUOSO Code Generator accomplishes this by structuring and standardizing the manner in which applications are developed. This type of structure and standardization provides many benefits: 1) higher quality and more consistent code; 2) lower maintenance costs, since maintenance programmers understand more quickly how to fix problems; and 3) less time spent in testing and debugging new applications, since standard, proven code is used. Programmers can use the VIRTUOSO Code Generator to essentially create their own high-level programming language which generates high performing, industry standard source code. They can actually build their own application generator!

Improving Development and Maintenance

The VIRTUOSO Code Generator increases programmer productivity for development as well as for ongoing application maintenance which often comprises 70-90% of programmers' time. Individuals typically develop code for their own use with little sharing or reuse of existing routines, leading to unproductive duplication of effort. The structure of the code, level of documentation, and naming conventions vary by programmer. These inconsistencies in style make it difficult to maintain code which has been developed by others.

The VIRTUOSO Code Generator can help solve these problems, leading to more productive programmers producing higher quality and more consistent work. This is accomplished by providing a framework for encouraging structured programming and standards. VIRTUOSO supports a structured programming methodology that provides for locally-defined standards and the reuse of existing code.

Using the VIRTUOSO Code Generator, all the programmers at a site will structure their code based on locally-defined models, and share code stored in code libraries. As a result, programming and documentation is more standardized and less affected by individual styles. Additionally, it is easier for new programmers placed on maintenance projects to become immediately productive. Also reduced is the time spent testing and debugging applications since standard, proven code is generated. Coding time is further reduced through the use of high level statements called macros which represent many lines of often used code.



Building Your Own Application Generator

Programmers can take advantage of all these benefits without having to learn a new programming language. They can build upon their existing knowledge of programming languages and application development.

VIRTUOSO With Existing Applications

While the benefits of the VIRTUOSO Code Generator may be clear for new applications, VIRTUOSO can be used to equal advantage in maintaining your existing applications. As these applications need revision, the macros you have chosen to use can be incorporated, making older applications easier to maintain over time. Also, you can leverage your existing applications by populating the VIRTUOSO Code Generator's reusable libraries with code from existing applications. Consequently, application systems developed using VIRTUOSO can coexist with current COBOL applications, databases and tools.

Key Features

- A customizable programming methodology provides a framework for setting local programming standards and results in more productive, higher quality development and maintenance.
- A source code generator allows applications to achieve high levels of performance using an industry standard third generation language.

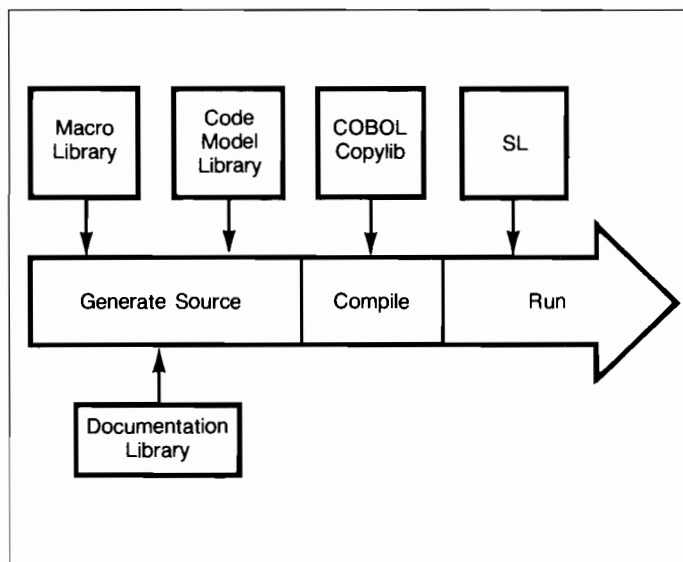
- High level macros, representing often used portions of code, can be created, modified, and shared, thus reducing the amount of programming required for new or changing applications.
- Model programs representing generic structures for program logic can also be created and shared.
- Central data definition in the System Dictionary gives programmers access to existing file structures and definitions. They are thus less likely to duplicate file structures and can more easily follow established data standards because they are aware that the information exists and is easily referenced.
- The ability to automatically generate program documentation from locally defined standards provides a means to keep up to date with modifications and establishes standardized documentation.

Functional Description

Each programming department will develop a set of macros and models which embody the locally-defined standards and practices. Individual programmers will compose specific programs by selecting from the department macro and model libraries. They will use the VIRTUOSO Code Generator to produce compilable source from the libraries of code.

Libraries of Reusable Code

The VIRTUOSO Code Generator uses libraries of reusable code that are accessed at generation time. The libraries facilitate standardization and code sharing among programmers. There are two types of libraries: macros and models. The macros form your own high level "pseudo code" language for performing standard operations. The models form reusable logic structures which implement common data processing solutions.



VIRTUOSO Code Generator Libraries

Faster Coding with Macros

In order to enhance programming productivity, the VIRTUOSO Code Generator enables programmers to use high level statements called macros. Macros can be designed for updating databases and managing screens, as well as for calling libraries of standard, commonly used routines. Each macro is expanded into a number of COBOL statements when the application is generated. Over 40 high level macros are supplied in a COBOL sample macro library. Some examples of supplied COBOL macros are those to open, read, write to, update and close a database.

Create Your Own Macros

Users can create their own macros to establish high-level statements for anything they wish to accomplish. If you want to make a change to something after a macro is created and source has been generated, you simply change the macro, regenerate, and recompile; you need not edit the generated source. Sample macros supplied with the VIRTUOSO Code Generator can be modified by users to meet local requirements. For instance, programmers can change the specifications or definitions of the supplied macros.

Easier Standardization with Models

In order to ease sharing and standardization, the VIRTUOSO Code Generator enables programmers to use modules which consist of model structures for their source code and for the structure and content of program documentation. A model is a generic structure that is defined according to local standards. The specific information which transforms a model into a unique program can be stored in the System Dictionary. The use of models helps enforce standards so that all code and documentation produced at a site look the same. This results in easier program maintenance. Sample models are supplied as part of the COBOL sample library. An example of a supplied model is a program structure to capture information from VPLUS screens to update an IMAGE dataset.

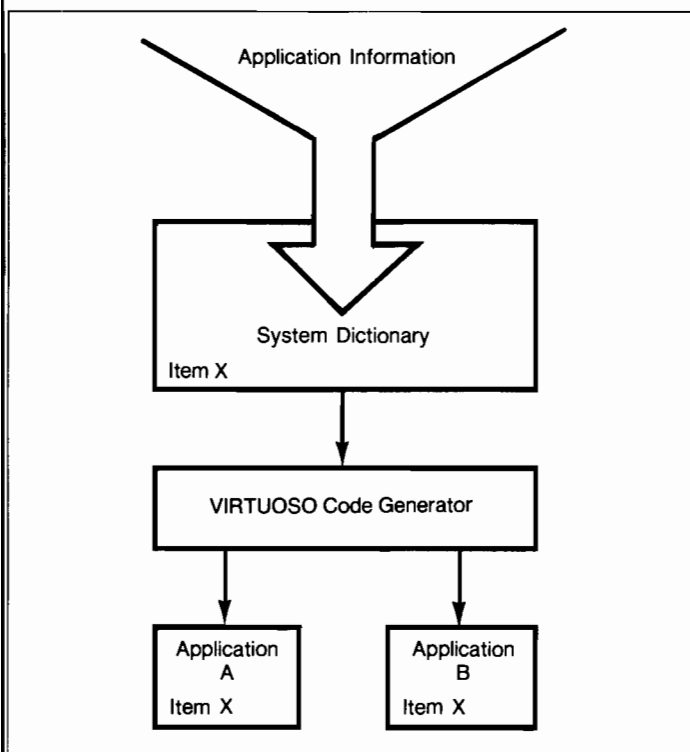
Sample Libraries

Sample macros and models are available which use COBOL. The code in the HP sample libraries is based on the 1985 ANSI COBOL standard. Customers can modify the samples, select some to use as is, or develop an entire new set of macros and models.

Solving Application Maintenance Problems

With traditional methods of program maintenance, changes made to applications require manually searching other applications to make corresponding changes. For instance, if the definition of a data item changes, the system must be searched to locate, isolate, and manually change the item at each location. This is a time consuming and unreliable process. Using the VIRTUOSO Code Generator, programmers can isolate often-changed application components and store them in the System Dictionary. This ability drastically reduces the amount of time programmers spend changing and updating applications, so the same amount of work can be accomplished by fewer people, and time can be freed up to develop new applications.

For example, if the same information appears in datasets which are used in an accounts receivable and an accounts payable program, then when the information changes, the programmer need only change it once in the programmer's place of choice. If the change is not made initially in the System Dictionary, it can then be registered in the System Dictionary. The VIRTUOSO Code Generator retrieves information about the applications from the System Dictionary and includes it in the generated source and program documentation. Consequently, if a change is necessary, it can be made once, and the applications can then be regenerated based on this changed information.



Using a Single Point of Change

The VIRTUOSO Programming Package

The VIRTUOSO Programming Package consists of several products which VIRTUOSO programmers will use to create and maintain applications.

When you order the VIRTUOSO Programming Package, you receive:

- VIRTUOSO Code Generator - for code and documentation generation
- VIRTUOSO COBOL Sample Library - contains macros and models to develop COBOL applications
- Business Report Writer - for report generation
- HP System Dictionary - for storing information about VIRTUOSO applications
- System Dictionary COBOL Definition Extractor - for generating copy members used by VIRTUOSO applications

There are several other tools you will use with the above products. The COBOL II (ANSI 85) compiler is used for source code compilation. Formspec (provided in the Fundamental Operating Software) is used for screen design. TurboIMAGE is the database management system supported by VIRTUOSO. And, any ASCII editor can be used within the VIRTUOSO Programming Package environment.

For further functional details on any of the products mentioned above, please see their respective data sheets in this Specification Guide.

System Environment

The VIRTUOSO Programming Package is supported on MPE V based HP 3000 systems.

The VIRTUOSO Programming Package requires the MPE V Fundamental Operating System.

Ordering Information

Software

Product

Product Number	Product Description
----------------	---------------------

30422A	VIRTUOSO Code Generator Right-to-Use Product
30422R	VIRTUOSO Code Generator Right-to-Copy with sublicense
30423A	VIRTUOSO Programming Package Right-To-Use Product includes: VIRTUOSO Code Generator VIRTUOSO COBOL Sample Library Business Report Writer HP System Dictionary System Dictionary COBOL Definition Extractor
30426A	VIRTUOSO COBOL Sample Library Right-to-Use Product

Documentation

Part

Part Number	Document Description
-------------	----------------------

30422-90001	VIRTUOSO Programming Package Reference Manual
-------------	--

Customer Training

A five-day course for using the VIRTUOSO Programming Package is available through the local sales office. The course includes instruction and labs, and is highly recommended for all purchasers of the VIRTUOSO Programming Package.

Course

Course Number	Course Name
---------------	-------------

30189	VIRTUOSO Programming Package User's Course
-------	---

HEWLETT-PACKARD

Product Number 32247A

Transact/V

For MPE V Based HP 3000 Systems

Transact — Increasing Programmer Productivity

Transact/V is a high-level programming language and interpreter for transaction processing applications. Designed as a procedural language, Transact/V provides the functionality of a third generation language, such as COBOL or Pascal, combined with a comprehensive set of powerful, high-level constructs that can perform several functions within a single statement.

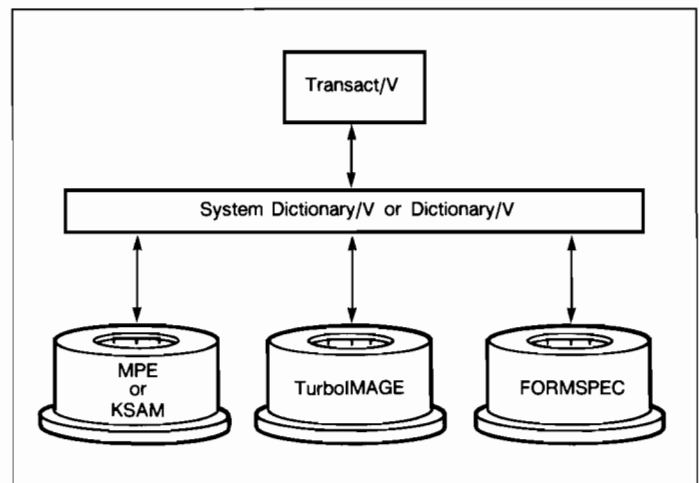
Applications written in Transact/V require fewer lines of code than those in traditional third generation languages. They are not only easier to write, they are also easier to understand and maintain. The result is significantly lower development and maintenance costs.

Integration with file management facilities is a major strength of Transact/V. Data definitions and locations can be accessed through System Dictionary/V or Dictionary/V instead of defining them individually in each program. Using Dictionary/V, data resolution can take place dynamically as the program is executing, providing total data independence; or at compile time for increased performance. When using System Dictionary/V, data resolution is accomplished at compile time. Access to TurboIMAGE/V data-bases, MPE V files, KSAM/V files, and FORMSPEC/V forms files supports the creation of a wide variety of transaction-oriented applications. It is also possible to access TurboIMAGE/XL databases and MPE XL files when running on 900 series HP 3000 computers.

Transact/V reduces program development and maintenance time, allowing programmers to develop information processing systems quickly, easily, and at a lower cost.

Key Transact/Features

- A consistent interface to various HP file structures means that programmers need only learn a single method of accessing TurboIMAGE/V and TurboIMAGE/XL data-bases, MPE V, MPE XL, and KSAM/V files, and FORMSPEC/V forms.
- A set of high-level statements (called verbs) for data access and manipulation perform bulk operations on behalf of the programmer.



- The ability to call HP 3000 system intrinsics offers extended capability to the application programmer.
- The ability to call run-time procedures written in other languages such as COBOL II/V, COBOL II/XL, Pascal/V, Pascal/XL, FORTRAN 77/V, FORTRAN 77/XL, and SPL/V, as well as in Transact/V permits sharing of commonly used routines.
- The ability to call Report/V, Inform/V, and Business Report Writer/V supports easy to design formatted output for reporting purposes.
- Automated control of Local Forms Storage off-loads the programmer from extensive programmatic screen control routines.
- Support of database and data set locking maintains the logical integrity of the data.
- A built-in command structure facilitates the creation of command-driven applications that will prompt the user for information.
- Automated resolution of data and file definitions through either System Dictionary/V or Dictionary/V eliminates the need for data definitions within the program.
- Both automatic and programmatic error handling and recovery support flexible error trapping.
- An extensive, built-in test facility aids the programmer in debugging Transact/V programs.

Functional Description

Transact/V is specifically designed for transaction processing applications. In fact, the "command-structure" feature relieves the programmer from much of the coding usually required to design menu-driven applications. The programmer assigns a label to a block of Transact/V statements that accomplish a given task. The label is then identified to the end user as a command name. When a command is entered in an interactive environment, the Transact/V processor executes the statements that have been defined for the given command. Using the commands, the end user controls the logical flow of the application each time it is executed. In addition to command structured applications, Transact/V can also be used to implement traditional screen-driven and batch-mode applications.

Transact/V consists of a high-level language, and interpreter. The compiler accepts source code and produces intermediate processor code. The processor then executes this code. Transact/V source code consists of very high-level statements with which the programmer can specify what actions are to take place. A set of registers, maintained by the processor, provide a consistent and logical interface to various file structures. Information held by the registers include:

- Data item definitions
- Data item values
- Keys and their values for keyed retrieval
- Match criteria for selection of specific entries
- Update values for easy data modification

Data definitions and structures can be maintained by either System Dictionary/V or Dictionary/V, or they can be defined within the Transact/V source code as with traditional programming languages.

Transact Functional Areas

Transact/V statements perform high-level functions in four major areas:

Database and file operation statements form a single interface to TurboIMAGE/V and TurboIMAGE/XL databases, MPE V, MPE XL, KSAM/V, and FORMSPEC/V files.

Data entry and display statements accept data input, automatically perform edit checking, and control the formatting and output of the data on the terminal screen.

Program control statements provide the logic and control the flow of the program, such as arithmetic operations, loops, branches, go to, and perform.

Assignment statements move data between items, compute expression values, set the various registers, and control global options such as page length and output device.

A single, high-level Transact/V statement will typically perform multiple operations that would require extensive coding if written in FORTRAN, COBOL, or Pascal.

Transact Applications

Transact/V provides the means to quickly develop a variety of applications. Batch-mode applications can be easily coded with the same statements used for interactive transaction processing. For interactive transaction processing applications, the built-in command structure eliminates the need to write code to handle commands. The command sequence can be implemented in block-mode or character-mode environments.

Block-mode, Interactive Systems

Transact/V has a high-level, integrated VPLUS/V interface to facilitate the use of block-mode applications. This interface dramatically reduces the amount of work necessary to communicate with VPLUS/V. For example, to add product information into an inventory database, a VPLUS/V interface would require the following Transact/V code:

```
##ADD:
$PRODUCT:
  list (auto) PRODUCT-INFO;
  get (form) PRODUCT-SCREEN, Init;
  put PRODUCT-INFO;
```

The GET statement handles the complete VPLUS/V interface, from opening the terminal and forms file through edit checking and data conversion. There is no need to specify any lower level intrinsics. Local form storage options are provided whenever possible.

Character-mode, Interactive Systems

Transact/V contains statements to accept data from the terminal, perform simple edit checking on the data entered, and then write the data to the appropriate file. Below is a set of statements that will prompt the end user for product information and then add the new information to the PRODUCT-INFO data set within INVTRY database:

```
##ADD:
$CUSTOMER:
  prompt PROD-NUM, checknot=PRODUCT-INFO:
  PROD-NAME:
  PROD-PRICE
  put PRODUCT-INFO,
  list=(PROD-NUM;PROD-PRICE);
```

The above sequence of statements will result in the following interaction at the terminal:

```
>add customer
PRODUCT NUMBER> 3266-4582
PRODUCT NAME> WIDGET
PRODUCT PRICE> 155
```

For each item, the user is prompted with an entry text as defined in System Dictionary/V or Dictionary/V. The programmer can define the entry text in the program if so desired. Basic edit checks are performed using the edit definitions stored in the dictionary. It is possible to continue the prompting, if necessary. At the end of each prompting sequence, the information is written to the PRODUCT-INFO data set within the INVTRY database. There is no need to call any system level intrinsics or, in this example, code any error checking routines.

Reporting Capabilities

In addition to transaction processing, Transact/V provides two methods for generating reports. Transact/V can call pre-defined Business Report Writer/V, Report/V, and Inform/V procedures, or reports can be defined within the Transact/V source. When defining reports within the source code, the headings and edit masks can be retrieved from either System Dictionary/V or Dictionary/V. This reduces the amount of work needed to be done by the programmer and ensures higher quality code. If desired, the programmer can also define the layout of the report within the Transact/V source code as well.

System Environment

Transact/V is supported on MPE V based HP 3000 systems (U-MIT or later) and in Compatibility Mode on the 900 Series of HP 3000 systems. Transact/V also requires TurboIMAGE/V or TurboIMAGE/XL; FORMSPEC/V; SORT/V; VPLUS/V; Language Intrinsics; System Dictionary/V or Dictionary/V; and system support of Native Language Support Intrinsics.

Ordering Information

The right to use Transact/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product	
Number	Product Description
32247A	Transact/V Right-to-Use product
32247R	Transact/V Right-to-Copy product with sublicense

Documentation

Part	
Number	Document Description
32247-90001	Transact Reference Manual
32247-90007	Getting Started with Transact

Customer Training

Hewlett-Packard offers both self-paced training and classroom instruction for Transact/V. Each course includes instruction and labs.

Course	
Number	Course Name
22842A	Programming in Transact
35014B	Introduction to Transact
51402A	Advanced Transact

Contact the local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Number 30138A

Transact/XL

For the 900 Series of HP 3000 Systems

Transact/XL — Increasing the Performance of Transact Applications

Transact/XL is a high-level programming language, interpreter, and compiler for transaction processing applications that execute on the 900 Series of HP 3000 systems.

Transact/XL consists of two main components: the Transact programming language and interpreter, and a Native Mode, object code compiler. Transact/XL provides the functionality and performance of a third generation language, such as COBOL or Pascal, combined with a comprehensive set of powerful, high-level constructs that can perform several functions within a single statement.

Applications written in Transact/XL require fewer lines of code than those in traditional third generation languages. They are not only easier to write, but they are also easier to understand and maintain. The result is significantly lower development and maintenance costs.

Integration with file management, database management, and screen management facilities is a major strength of Transact/XL. Data definitions and locations can be accessed through System Dictionary/XL or Dictionary/V instead of defining them individually in each program. Also, access to TurboIMAGE/XL, MPE/XL, KSAM/V, and FORMSPEC/V forms files supports the creation of a wide variety of transaction oriented applications.

This data sheet provides information on the Transact/XL compiler. For a functional description of the Transact language, please refer to the Transact/V data sheet included in this specification guide.

Transact/XL Compiler

The Transact/XL compiler provides users with a means by which they can run their Transact applications in Native Mode on the 900 Series of HP 3000 systems. Unlike interpreted statements, compiled Transact statements provide the user with a performance level comparable to that of traditional third generation languages, like COBOL and Pascal.

Key Features

- Automated resolution of data and file definitions and locations at compile time through System Dictionary/XL or Dictionary/V eliminates data definitions within the program.
- Both automatic and programmatic error handling and recovery supports flexible error trapping.
- The ability to call HP 3000 system intrinsics offers extended capability to the application programmer.
- The ability to call run-time procedures written in other languages such as COBOL II/XL, Pascal/XL, and FORTRAN 77/XL, as well as in Transact, permits sharing of commonly used routines.
- The ability to call Business Report Writer/XL supports easy to design formatted output for reporting purposes.

System Environment

Transact/XL is supported on the 900 Series of HP 3000 systems.



Ordering Information

The right to use Transact/XL on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product

Number	Product Description
30138A	Transact/XL Right-to-Use Product
30138R	Transact/XL Right-to-Copy Product with sublicense

Documentation

Documentation for Transact/XL is included in the documents available for Transact/V.

Product

Number	Document Description
32247-90001	Transact Reference Manual
32247-90007	Getting Started with Transact

Customer Training

Training for Transact/XL is included in the training available for Transact/V. Hewlett-Packard offers both self-paced training and classroom instruction for these products. Each course includes instruction and labs.

Course

Number	Course Name
22842A	Programming in Transact
35014B	Introduction to Transact
51402A	Advanced Transact

Contact the local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Number 32350A

Toolset/V

For MPE V Based HP 3000 Systems

Maximizing Programmer Productivity

Toolset/V provides a set of tools that work together to maximize the productivity of COBOL II/V and Pascal/V programmers in the coding, symbolic debugging, and version management phases of development on the HP 3000; and extends symbolic debugging to include support for FORTRAN 77/V. These tools perform many of the fundamental and routine programming tasks and simplify or facilitate others.

Key Toolset/V Features

- Workspace File Manager for COBOL II/V and Pascal/V
- Friendly and Powerful User Interface
- Full Screen Editor for COBOL II/V and Pascal/V
- COBOL II/V COPY Library Editing and Management
- COBOL II/V and Pascal/V Data Definition Retrieval from Dictionary/V
- Program Translation Manager for COBOL II/V and Pascal/V
- Symbolic Debug for COBOL II/V, Pascal/V, and FORTRAN 77/V
- Help Facility

Functional Description

Source entry, version management, and debugging are all combined into one set of tools that work together to increase programmer productivity.

Workspace File Manager

The Workspace File Manager maintains all files needed for the development of a program — the source files, versions of source files, \$INCLUDE files, USL file, compiler listing files, and program file.

It also manages:

- COBOL COPY libraries
- Labeling documentation for each version of a source file
- Sharing of source file information between members of a programming team.
- Storing only the differences between versions to improve disc utilization.
- Listing changes between versions to simplify code maintenance.

The Workspace File Manager features a file storage format that allows all source files and versions of source files within a workspace to be stored in a special data format using data compression techniques. A simple command enables the programmer to convert existing COBOL II and Pascal source files into this special data format providing a more efficient and cost-effective utilization of disc storage.

User Interface

The User Interface allows the programmer to use either simple, easy-to-use, screen-labeled function keys and menus or to directly key in commands.

Full Screen Editor

The features of the Full Screen Editor are:

- Full range direct screen editing for simple, easy entry and modification of source code within a Toolset file or a COBOL COPY library module
- Screen-labeled function keys for adding, moving, and changing text; locating and changing character strings; and browsing through source files and compile listings
- Terminal editing keys for inserting and deleting characters and lines
- Automatic source file updating

COBOL Copy Library Keys and Commands

The COBOL COPY Library keys and commands allow the programmer to easily create, edit, and copy COBOL COPY libraries and to list their contents.

Dictionary Data Definition Retrieval

In addition, the GENERATE command in Toolset accesses the COBOL II/V and Pascal/V Definition Extract Utilities which are components of Dictionary/V. These utilities extract data definitions from Dictionary/V and use them to generate COBOL and Pascal data declarations.

COBOL data declarations can be generated for the Environment Division and the Working Storage and File sections of the Data Division.

Further, many COBOL clauses can also be generated, such as the "RECORD KEY IS" clause for the Environment Division and the "BLOCK CONTAINS" clause for the Data Division. The generated code is put into a Toolset editor file or a COBOL COPY Library. Pascal data definitions can be generated for Pascal VAR and TYPE declarations; the generated code is put into a Toolset editor file.

Program Translation Manager

The Program Translation Manager consists of screen-labeled function keys which allow programmers to compile, prepare, and run programs with minimum effort and maximum effectiveness. The Program Translation Manager also supports on-line storage of the latest compile listing and availability of MPE V :PREP and :RUN commands.

The screen labeled program keys and their functions:

- Program function keys to compile, prepare, or run a COBOL II or Pascal source program
- GO key to compile, prepare, and run a source program in a single step
- LISTING key to display the compiler listing on-screen
- FIND ERR key to locate errors in the compiler listing
- EDIT key to display program lines for easy editing of the source file while you are debugging

Other features of the Program Translation Manager:

- On-line storage of the latest compile listing
- Availability of MPE V :PREP and :RUN commands

Symbolic Debug

Symbolic Debug is a facility that speeds interactive program debugging at the source code level.

It is simple, fast, easy to learn, and alleviates the need to understand the internals of the system in order to debug COBOL II, FORTRAN 77, and Pascal programs.

Symbolic debug has the following key functions:

- References variables symbolically using their names in the source code
- Sets and Clears program breakpoints with optional frequency and duration counts
- Displays values and moves new values into any data items
- Trace and Retrace program flow
- Trace changes in data item values between COBOL paragraphs, FORTRAN subroutines, and Pascal procedures

In addition for COBOL II and Pascal, Symbolic Debug allows the programmer to:

- Edit a source file at a breakpoint
- Display a compile listing at a breakpoint

Programmers can also use the Symbolic Debug utility for COBOL II, FORTRAN 77, and Pascal programs which were created and modified outside of Toolset.

Help Facility

The Help Facility provides:

- On-line information on commands, syntax, and other functions
- Cause and corrective information for errors
- Context sensitive information in that at the time the error occurs the user can bring up the help facility for more specific information

System Environment

Toolset/V is supported on MPE V based HP 3000 systems.

Toolset/V requires at least one of the following compilers: COBOL II/V, HP FORTRAN 77/V, or Pascal/V. Further, Toolset/V requires VPLUS/V and VPLUS-compatible terminals to fully support screen functionality.

Ordering Information

The right to use Toolset/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
32350A	Toolset/V Right-to-Use product
32350R	Toolset/V Right-to-Copy product with sublicense
31503A	COBOL II Developer's Package (COBOL II/V & Toolset/V) Right-to-Use product
31503R	COBOL II Developer's Package Right-to-Copy product with sublicense
31504A	FORTRAN 77 Developer's Package (HP FORTRAN 77/V & Toolset/V) Right-to-Use product
31504R	FORTRAN 77 Developer's Package Right-to-Copy product with sublicense
31505A	Pascal Developer's Package (Pascal/V & Toolset/V) Right-to-Use product
31505R	Pascal Developer's Package Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
32350-90001	Toolset User's Manual

Customer Training

Hewlett-Packard offers a self paced training course for Toolset including instruction and labs.

Course Number	Course Name
32350-90002	Toolset Self-Paced Training Course

The on-line help facility provided with Toolset, along with the Toolset menu screens, serves as an additional guide to learning the product.

HEWLETT-PACKARD

Product Number 36044A

Toolset/XL

For the 900 Series of HP 3000 Systems

Maximizing Programmer Productivity

Toolset/XL provides a set of tools that work together to maximize the productivity of COBOL II/XL and Pascal/XL programmers in the coding, symbolic debugging, and version management phases of development on the 900 Series of HP 3000 systems; and extends symbolic debugging to include support for HP FORTRAN 77/XL. These tools perform many of the fundamental and routine programming tasks and simplify or facilitate others.

Key Toolset/XL Features

- Workspace File Manager for COBOL II/XL and Pascal/XL
- Friendly and Powerful User Interface
- Full Screen Editor for COBOL II/XL and Pascal/XL
- COBOL II/XL COPY Library Editing and Management
- COBOL II/XL and Pascal/XL Data Definition Retrieval from Dictionary/V
- Program Translation Manager for COBOL II/XL and Pascal/XL
- Symbolic Debug for COBOL II/XL, Pascal/XL, and HP FORTRAN 77/XL
- Help Facility

Functional Description

Source entry, version management, and debugging are all combined into one set of tools that work together to increase programmer productivity.

Workspace File Manager

The Workspace File Manager maintains all files needed for the development of a program — the source files, versions of source files, \$INCLUDE files, RL file, compiler listing files, and program file.

It also manages:

- COBOL COPY libraries
- Labeling documentation for each version of a source file
- Sharing of source file information between members of a programming team
- Storing only the differences between versions to improve disc utilization
- Listing changes between versions to simplify code maintenance

The Workspace File Manager features a file storage format that allows all source files and versions of source files within a workspace to be stored in a special data format using data compression techniques. A simple command enables the programmer to convert existing COBOL II and Pascal source files into this special data format providing a more efficient and cost-effective utilization of disc storage.

User Interface

The User Interface allows the programmer to use either simple, easy-to-use, screen-labeled function keys and menus, or to directly key in commands.

Full Screen Editor

The features of the Full Screen Editor are:

- Full range direct screen editing for simple, easy entry and modification of source code within a Toolset file or a COBOL COPY library module.
- Screen-labeled function keys for adding, moving, and changing text; locating and changing character strings; and browsing through source files and compile listings.
- Terminal editing keys for inserting and deleting characters and lines.
- Automatic source file updating.

COBOL Copy Library Keys and Commands

The COBOL COPY Library keys and commands allow the programmer to easily create, edit, and copy COBOL COPY libraries and to list their contents.

Dictionary Data Definition Retrieval

The GENERATE command in Toolset accesses the COBOL II/XL and Pascal/XL Definition Extract Utilities which are components of Dictionary/V. These utilities extract data definitions from Dictionary/V and use them to generate COBOL and Pascal data declarations.

COBOL data declarations can be generated for the Environment Division and the Working Storage and File sections of the Data Division.

Further, many COBOL clauses can also be generated, such as the "RECORD KEY IS" clause for the Environment Division and the "BLOCK CONTAINS" clause for the Data Division. The generated code is put into a Toolset editor file or a COBOL COPY Library. Pascal data definitions can be generated for Pascal VAR and TYPE declarations; the generated code is put into a Toolset editor file.

Program Translation Manager

The Program Translation Manager consists of screen labeled function keys which allow programmers to compile, link, and run programs with minimum effort and maximum effectiveness. The Program Translation Manager also supports on-line storage of the latest compile listing and availability of MPE XL :LINK and :RUN commands.

The screen-labeled program keys and their functions:

- Program function keys to compile, link, or run a COBOL II or Pascal source program
- GO key to compile, link, and run a source program in a single step
- LISTING key to display the compiler listing on-screen
- FIND ERR key to locate errors in the compiler listing
- EDIT key to display program lines for easy editing of the source file while you are debugging

Other features of the Program Translation Manager:

- On-line storage of the latest compile listing
- Availability of MPE XL :LINK and :RUN commands

Symbolic Debug

Symbolic Debug is a facility that speeds interactive program debugging at the source code level.

It is simple, fast, easy to learn, and alleviates the need to understand the internals of the system in order to debug COBOL II, FORTRAN 77, and Pascal programs.

Symbolic debug has the following key functions:

- References variables symbolically using their names in the source code
- Sets and Clears program breakpoints with optional frequency and duration counts
- Displays values and moves new values into any data items
- Trace and Retrace program flow
- Trace changes in data item values between COBOL paragraphs, FORTRAN subroutines, and Pascal procedures

In addition for COBOL II and Pascal, Symbolic Debug allows the programmer to:

- Edit a source file at a breakpoint
- Display a compile listing at a breakpoint

Programmers can also use the Symbolic Debug utility for COBOL II, FORTRAN 77, and Pascal programs which were created and modified outside of Toolset.

Help Facility

The Help Facility provides:

- On-line information on commands, syntax, and other functions
- Cause and corrective information for errors
- Context sensitive information in that at the time the error occurs the user can bring up the help facility for more specific information

System Environment

Toolset/XL is supported on the 900 Series of HP 3000 systems.

Toolset/XL requires at least one of the following compilers: COBOL II/XL, HP FORTRAN 77/XL, or Pascal/XL. Further, Toolset/XL requires VPLUS/V and VPLUS-compatible terminals to fully support screen functionality.

Ordering Information

The right to use Toolset/XL on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
36044A	Toolset/XL Right-to-Use product
36044R	Toolset/XL Right-to-Copy product with sublicense
31503A	COBOL II Developer's Package (COBOL II/XL & Toolset/XL) Right-to-Use product
31503R	COBOL II Developer's Package Right-to-Copy product with sublicense
31504A	FORTRAN 77 Developer's Package (HP FORTRAN 77/XL & Toolset/XL) Right-to-Use product
31504R	FORTRAN 77 Developer's Package Right-to-Copy product with sublicense
31505A	Pascal Developer's Package (Pascal/XL & Toolset/XL) Right-to-Use product
31505R	Pascal Developer's Package Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
32350-90001	Toolset User's Manual

Customer Training

Hewlett-Packard offers a self-paced training course for Toolset including instruction and labs.

Course Number	Course Name
32350-90002	Toolset Self-Paced Training Course

The on-line help facility provided with Toolset, along with the Toolset menu screens, serves as an additional guide to learning the product.

HP's Screen Management Facility

For MPE V Based and 900 Series of HP 3000 Systems

Powerful Screen Management Services

HP's screen management facility provides powerful, user friendly services that simplify both the creation of forms and the development of terminal I/O portions of applications. These services consist of three primary components: FORMSPEC/V, High Level Screen Management Intrinsic Library (Hi-Li) and VPLUS/V. All are included as part of the Fundamental Operation Software (FOS).

Overview

FORMSPEC/V is an easy to use, menu driven, interactive form design facility for applications programmers. The form developed can then be manipulated by applications using either the High Level Screen Management Intrinsic Library (Hi-Li) or VPLUS/V intrinsics. Additionally separating the screen interface from the application allows maintenance of the screen image to occur without changes to the application.

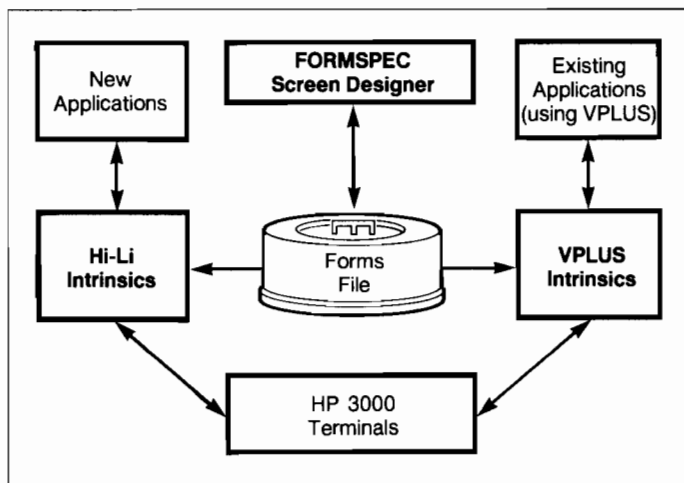
Both Hi-Li and VPLUS/V consist of a set of intrinsics that programmers can use to develop the terminal I/O portions of applications. Hi-Li is a powerful set of high-level user interface intrinsics designed to simplify application development by combining logical sets of low-level terminal I/O operations into macro-like intrinsics. Hi-Li greatly simplifies a programmer's activities by taking the functions represented in approximately 55 VPLUS/V intrinsics and condensing them into 12. The result is fewer intrinsics to learn, code, and maintain.

When To Use Hi-Li

With the increased efficiency gains in programmer productivity in both the coding and maintenance phase of applications, most programmers will select Hi-Li when developing new applications. Note that Hi-Li and VPLUS/V intrinsics may not both be called from the same application.

When to Use VPLUS/V

Programmers maintaining existing applications that call VPLUS/V intrinsics will continue to support those applications using VPLUS/V. The execution performance of Hi-Li and VPLUS/V is equivalent, therefore, recoding existing VPLUS/V applications for performance gains is not recommended.



Key FORMSPEC/V Features

- Provides an easy to use, menu driven forms design facility.
- Supports the creation of forms to be used on block-mode terminals available from Hewlett-Packard.
- Supports the creation of forms that utilize the various special features of HP workstations.
- Provides capability to define the data format for each field in a form.
- Supports comprehensive data editing and validation.
- Supports arithmetic and conditional processing.
- Supports custom run-time error messages.

Resulting Benefits of FORMSPEC/V

FORMSPEC/V provides programmers with a facility to design screens for creating a friendly interface for their applications. Programmers can take advantage of the powerful features of Hewlett-Packard workstations and minimize the time necessary for design and coding. Programmer productivity is enhanced by simplifying the interactive design of data entry screens. Users of resulting applications are more productive because they work with an application that is easy to learn and use.

FORMSPEC/V Functional Description

User Friendly, Forms Design Facility

With the FORMSPEC/V forms design facility, programmers can interactively develop screens for applications, taking advantage of the comprehensive data editing capabilities.

Application screen formats are laid out interactively on a terminal screen. Each form contains fields with characteristics defined from a set of standard descriptions such as type of field (e.g., required, optional, or display only) and data type (e.g., character, numeric, or date). Default values can be provided for each field. The programmer can use FORMSPEC/V to define the following functions instead of coding them independently in the program:

- Comprehensive data editing capabilities include length checking, range checking, list checking, equality checks, pattern matching, and check digit verification (modulo 10 or 11).
- Data can be formatted as it is being collected. Standard routines are available that justify, fill, strip, and upshift data in the fields specified.
- Data movement can be specified to transfer values between fields in a single form or field values between forms. For example, this allows the sum of several fields in a form to be moved to another field in the same form reserved for the total.
- Arithmetic and conditional processing, dependent on the value entered in a field, can be defined and specified as needed.
- Custom error messages for display at run-time can be specified with each error characterization.
- Definition of the data editing and final format rules for the Application Ready Buffer (ARB) is supported by Hi-Li and VPLUS/V Intrinsics. Through use of the Application Ready Buffer (ARB), programmers can specify the data editing rules and final format for all the data fields within a form. Hi-Li or VPLUS/V will then handle the entire form display, data entry, validation, and transformation process before returning to the application with the data, ready to use.
- Multiple screens can be linked together, and the sequence in which forms are presented can be controlled during data collection.

Stand-Alone Screen Testing Facility

The program, ENTRY, is provided for situations that require a stand-alone data entry capability. When ENTRY is run it accesses forms created with FORMSPEC/V. As the data is entered by the user, the ENTRY program utilizes the routines specified by the designer for each field and writes the data to an MPE batch file for later processing. Additionally, ENTRY can be used as a prototyping tool to test the designed screens without having to write an application program.

Hi-Li and VPLUS/V Feature Comparison

The functions of Hi-Li compare almost identically to VPLUS/V. The exceptions are:

- Hi-Li does not support terminal caching or batch I/O.
- Hi-Li does not support HP264x or HP's industrial terminals (i.e., HP30xx).

Hi-Li may be thought of as a concentrated superset of VPLUS/V. It offers the same functions but packs more power into a single intrinsic. The resulting advantage Hi-Li holds over VPLUS/V is greater programmer productivity. The features, benefits and functional description below are presented in the context of Hi-Li to simplify the explanation and eliminate redundancy.

Key Hi-Li Features

- Provides a high-level, macro-like intrinsic interface to HI programming languages.
- Supports block-mode terminals available from Hewlett-Packard.
- Supports user interface features such as HPTouch and color.
- Supports comprehensive data editing and validation.
- Supports automatic data transformation with the Application Ready Buffer (see Automatic Data Transfer section).

Resulting Benefits of Hi-Li

The High Level Screen Management Intrinsic Library (Hi-Li) helps programmers present a friendly, easy-to-use interface for the applications they develop. Hi-Li combines the most commonly used logical sets of low-level terminal I/O procedures into high-level intrinsic calls. This increases programmer productivity by simplifying the application development process. Application maintenance time is decreased because the application is not affected by system-level changes in terminal handling software.

The intrinsics also reduce the number of transfers between the application and user interface facilities. This increases software efficiency reduces error handling requirements, and simplifies application documentation and support as each intrinsic performs several system-level functions. Hi-Li also protects the programmer from hardware changes because applications are developed using the high-level intrinsics.

Hi-Li Functional Description

Simplified Forms Management

Hi-Li provides the capability to:

- Configure the terminal workstation.
- Retrieve a form definition from a forms file.
- Display a form with application provided data and window message and enhanced fields.
- Returns data that is entered into a displayed form along with the value of the last key pressed to the application.
- Execute logical groups of systems level terminal management routines with high-level intrinsics.
- Perform diagnostic activities during program development and debugging.

Workstation Support

Hi-Li and VPLUS/V support the features of all Hewlett-Packard block-mode workstations. This means, special workstation features can be utilized to tailor the user interface to the specific task. The following special features are supported:

- Color display.
- HPTouch.
- User definable labeling keys.
- Highlight, flashing, underline, half-bright, and security display enhancements.
- Local forms cache (VPLUS only).
- Modified data tag.

High Level Language Support

Hi-Li and VPLUS/V provide intrinsics which give the programmer specific control over the functions needed in the application's user interface. These intrinsics are available for programmers using BASIC/V, COBOL II/V, FORTRAN 66/V, FORTRAN 77/V, Pascal/V, and SPL/V. Additionally, the intrinsics are accessible by programs developed in COBOL II/XL, HP C/XL, HP FORTRAN 77/XL and HP Pascal/XL on the 900 Series of HP 3000 systems (see System Environment and Usage Considerations sections).

Usage Considerations

- VPLUS/V and Hi-Li intrinsics may not both be called from the same application.

System Environment

- High Level Screen Management Intrinsic Library (Hi-Li) is supported on MPE V and the 900 Series of HP 3000 systems.
- FORMSPEC/V and VPLUS/V are supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems. Additionally, VPLUS/V may be called transparently on MPE XL.
- Hi-Li and VPLUS/V require FORMSPEC/V for forms design, display, and data editing features.

Ordering Information

The right to use FORMSPEC/V, Hi-Li and VPLUS/V on an HP 3000 system is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

FORMSPEC/V, Hi-Li, and VPLUS/V are bundled with the Fundamental Operating Software (FOS) on all HP 3000 computer systems.

Documentation

Part Number	Documentation Description
32209-90001	VPLUS/V Reference Manual
32424-90001	High Level Screen Management Intrinsic Library Reference Manual

Customer Training

Hewlett-Packard offers classroom and self-paced instruction for FORMSPEC/V as part of VPLUS/V training.

Course Number	Course Name
32209-90004	Using VPLUS/V: An Introduction to Forms Design (self-paced)
22830C	VPLUS/V for Programmers
22830D	VPLUS/V For Designers (FORMSPEC/V usage)

Contact your local sales office for a course outline and schedule.

HEWLETT-PACKARD

Product Number 32233A

COBOL II/V

For MPE V Based HP 3000 Systems

COBOL II/V is a high-level language for business data processing that operates under the MPE V operating system. It takes full advantage of the system facilities and the MPE V stack architecture. COBOL II/V is based on the 1985 ANSI COBOL Standard X3.23-1985.

Key COBOL II/V Features

Standards Compliance

COBOL II/V conforms to the intermediate level of the ANSI COBOL X3.23-1985 specifications. It is designed to meet the corresponding level requirements of the United States government as described in FIPS PUB 21-1. Some of the individual modules are designed to meet the high-level requirements, as described below:

ANSI 85 Module	Level Supporting COBOL II	FIPS PUB21-1 Reqmt's For High Level
NUCLEUS	1	2
SEQUENTIAL I/O	1	2
RELATIVE I/O	2	2
INDEXED I/O	2	2
SEGMENTATION	2	2
SOURCE TEXT MANIPULATION	2	2
DEBUG*	1	2
INTERPROGRAM COMMUNICATION	1	2
SORT/MERGE	1	1
COMMUNICATION*	—	2
REPORT WRITER*	—	2

*Represents an optional module.

- **NUCLEUS MODULE:** Defines structures for elementary internal processing of data within the basic structure of the four divisions of a COBOL program. (Meets Level-2 implementation except for the INSPECT TALLYING ALL/LEADING list syntax.)
- **SEQUENTIAL I/O MODULE:** Provides the capability for defining and accessing sequentially organized files. (Meets Level-2 implementation except padding character, and record delimiter clauses as defined by the '85 standard.)
- **RELATIVE I/O MODULE:** Provides the capability for defining and accessing files in which records are identified by relative record number.

- **INDEXED I/O MODULE:** Provides the capability of defining and accessing files in which records are identified by key values as well as sequentially and randomly within the same COBOL program.
- **SEGMENTATION MODULE:** Provides for segmentation of Procedure Division sections at object time.
- **SOURCE TEXT MANIPULATION:** Provides for insertion into a program of predefined COBOL text.
- **DEBUG:** Allows a trace of the execution of a program in order to determine logic errors.
- **INTER-PROGRAM COMMUNICATION MODULE:** Allows COBOL to transfer control to one or more sub-programs which may be written in Hewlett-Packard's COBOL, FORTRAN, SPL or Pascal compiler languages, and whose names may or may not be known at compile time. (Meets Level-2 implementation except for nested source programs, batched compilation, and GLOBAL data.)
- **SORT-MERGE MODULE:** Provides sorting and merging capabilities of single and multiple input files. ASCII and EBCDIC data types and user defined collating sequences are implemented to allow the tailoring of the SORT-MERGE capability to specific data and sequence requirements of the particular application.

COBOL II includes the following new features of the X3.23-1985 ANSI COBOL standard:

- EVALUATE statement
- In-line PERFORM
- Scope-delimited statements
- CONTINUE statement
- Optional FILLER clause
- INITIALIZE statement
- Reference modification
- De-edited MOVE operations
- BY CONTENT argument passing
- Symbolic characters
- CLASS clause
- SET statement
- NOT phrases
- REPLACE statement
- Alphabetic tests
- External data and files

Language Extensions

Hewlett-Packard has implemented several extensions to the standard in order to increase the capability of COBOL II/V. The most significant extensions are:

- **PRE-PROCESSOR FUNCTION** provides statements that allow the programmer to equate a particular "macro" or sequence of code (\$DEFINE) or a file (\$INCLUDE) to an identifier. This identifier can be referenced throughout the program, and at compilation it is replaced with the expanded source code or source program file that it represents.
- **PROGRAM DEBUGGING AIDS** allow for the increased efficiency of the COBOL programming staff by providing tools that aid in finding problems within a program during the development cycle.
- **CROSSREF** provides a listing of all symbols and tables used in the expanded source file.
- **VERB MAP** provides a listing of COBOL statements and their location within the object code.
- **MPE DEBUG** allows a programmer to interactively trace through the execution of a program in order to determine logic errors.
- **MICROCODED INSTRUCTIONS** allow frequently utilized COBOL instructions such as the PERFORM statement, the EDIT instruction, and extended DECIMAL instructions to be coded into system firmware to improve the run-time efficiency of application programs.

Access to Subprograms

COBOL II/V provides the capability to call subprograms written in Hewlett-Packard's COBOL, FORTRAN, Pascal, and SPL. This allows for the usage of pre-existing code to perform a desired function.

Access to MPE V System Intrinsic

Allows easy programmatic access to the power of the MPE V operating system through calls to the system intrinsic.

- Allows for passing parameters by value
- Allows for passing byte and word address parameters and usage of decimal or octal literals
- Allows checking of Condition Codes indicating the status of execution of the intrinsic called

Accept Free Phrase

Allows a free field format for low-volume data entry.

File Locking Capability

Permits a program to request exclusive access to an MPE V or KSAM/V file through the usage of the EXCLUSIVE and UN-EXCLUSIVE statements.

Special Registers

TALLY, CURRENT-DATE, TIME-OF-DAY and WHEN-COMPILED are constructs frequently used programmatically and provided by COBOL II/V to reduce programming effort by eliminating the need to construct these registers within the WORKING-STORAGE SECTION of the program.

Packed Decimal (COMP-3)

Allows for efficient storage of data and performance efficiencies using decimal arithmetic for computations.

Multiple Entry Points to Subprograms

Reduces code redundancy by allowing similar functions within identical portions of code to be coded into a single subprogram and to be called through multiple names.

Data Management Interface

COBOL II provides:

- Access to both sequential MPE V and indexed sequential (KSAM/V) files through the usage of ANSI Standard COBOL Input and Output operations
- Access to Hewlett-Packard's VPLUS/V (screen management), TurboIMAGE/V and HPSQL/V (data management) and DSG (graphics) products through the usage of procedure libraries
- Access from HPToolset which includes an editor, workspace manager the ability to compile, prep, and execute with debug and interactive symbolic debugging
- Access to data definitions stored in Dictionary/V through HPToolset/V, allowing for the generation of COBOL Environment and Data Division entries from the dictionary, automatically and concurrently with program development
- The COBOL Definition Extractor utility is available to generate definitions for COBOL copylibs from System Dictionary/V definitions

Compatibility

The COBOL II/V compiler offers compatibility with the ANSI COBOL'74 Standard. Through the use of dual points of entry into the compiler, customers may choose which standard to execute, thereby, eliminating any need to convert between the two standards. The dual points of entry allow customers to avoid the few existing incompatibilities between the COBOL'74 and the COBOL'85 standards (e.g., reserved words), and thereby permit programmers to develop new code using the '85 standard while preserving their existing source code. Or if customers choose to move all their applications to the COBOL'85 Standard, the compiler will flag the syntax changes allowing customer to modify, recompile and execute using the ANSI COBOL'85 entry point. Code compiled with the ANSI COBOL'85 standard will run on MPE V/R; MPE V/E T-Delta-4, 5, and 6 MITS; and on UB-MIT and forward.

System Environment

COBOL II/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product

Number	Product Description
32233A	COBOL II/V Right-to-Use product
32233R	COBOL II/V Right-to-Copy product with sublicense

Documentation

Part

Number	Document Description
32233-90001	COBOL II/V Reference Manual

Customer Training

Course

Number	Course Name
22832A	Self-Paced Training, Learning COBOL II



HEWLETT-PACKARD

Product Number 31500A

COBOL II/XL

For the 900 Series of HP 3000 Systems

COBOL II/XL is a high-level language for business data processing that operates on the 900 Series of HP 3000 systems, taking full advantage of the system facilities and the HP Precision Architecture. It is based on the 1985 ANSI COBOL Standard X3.23-1985.

Key COBOL II/XL Features

Standards Compliance

COBOL II/XL conforms to the intermediate level implementation defined by the ANSI COBOL X3.23-1985 specifications. It is designed to meet the corresponding level requirements of the United States government as described in FIPS PUB 21-1. Some of the individual modules are designed to meet the high-level requirements, as described below:

ANSI 85 Module	Level Supporting COBOL II	FIPS PUB21-1 Reqmt's For High Level
NUCLEUS	1	2
SEQUENTIAL I/O	1	2
RELATIVE I/O	2	2
INDEXED I/O	2	2
SEGMENTATION	2	2
SOURCE TEXT MANIPULATION	2	2
DEBUG*	1	2
INTERPROGRAM COMMUNICATION	1	2
SORT/MERGE	1	1
COMMUNICATION*	—	2
REPORT WRITER*	—	2

*Represents an optional module.

- **NUCLEUS MODULE:** Defines structures for elementary internal processing of data within the basic structure of the four divisions of a COBOL program. (Meets Level-2 implementation except for the INSPECT TALLYING ALL/LEADING list syntax.)
- **SEQUENTIAL I/O MODULE:** Provides the capability for defining and accessing sequentially organized files. (Meets Level-2 implementation except for padding character and record delimiter clauses as defined by the '85 standard.)
- **RELATIVE I/O MODULE:** Provides the capability for defining and accessing files in which records are identified by relative record number.

- **INDEXED I/O MODULE:** Provides the capability of defining and accessing files in which records are identified by key values as well as sequentially and randomly within the same COBOL program.
- **SEGMENTATION MODULE:** Implements the rules for ALTERable GOTO's only.
- **SOURCE TEXT MANIPULATION:** Provides for insertion into a program of predefined COBOL text.
- **DEBUG:** Allows a trace of the execution of a program in order to determine logic errors.
- **INTER-PROGRAM COMMUNICATION MODULE:** Allows COBOL to transfer control to one or more sub-programs that may be written in Hewlett-Packard's COBOL, FORTRAN, Pascal and SPL compiler languages and whose names may or may not be known at compile time. (Meets Level-2 implementation except for nested source programs, batched compilation, and EXTERNAL & GLOBAL data.)
- **SORT-MERGE MODULE:** Provides sorting and merging capabilities of single and multiple input files. ASCII and EBCDIC data types and user defined collating sequences are implemented to allow the tailoring of the SORT-MERGE capability to specific data and sequence requirements of the particular application.

COBOL II/XL includes the following new features of the X3.23-1985 ANSI COBOL Standard:

- EVALUATE statement
- In-line PERFORM
- Scope-delimited statements
- CONTINUE statement
- Optional FILLER clause
- INITIALIZE statement
- Reference modification
- De-edited MOVE operations
- BY CONTENT argument passing
- Symbolic characters
- CLASS clause
- SET statement
- NOT phrases
- REPLACE statement
- Alphabetic tests
- External data and files

Language Extensions

Hewlett-Packard has implemented several extensions to the Standard in order to increase the capability of COBOL II/XL. The most significant extensions are:

- **PRE-PROCESSOR FUNCTION** provides statements that allow the programmer to equate a particular "macro" or sequence of code (\$DEFINE) or a file (\$INCLUDE) to an identifier. This identifier can be referenced throughout the program and is replaced at compilation with the expanded source code or source programs file that it represents.
- **PROGRAM DEBUGGING AIDS** allows for the increased efficiency of the COBOL programming staff by providing tools that aid in finding problems within a program during the development cycle.
- **CROSSREF** provides a listing of all symbols and tables and their locations within the source code.
- **VERB MAP** provides a listing of COBOL statements and their location within the object code.
- **MPE DEBUG** allows a programmer to interactively trace through the execution of a program in order to determine logic errors.

Access to Subprograms

COBOL II/XL provides the capability to call subprograms written in Hewlett-Packard's COBOL, FORTRAN, Pascal and SPL compiler languages. This allows for the usage of existing code.

Access to MPE XL System Intrinsic

Allows easy programmatic access to the power of the MPE XL operating system through direct access to the system intrinsic.

- Allows for passing parameters by value
- Allows for usage of decimal or octal literals
- Allows checking of Condition Codes indicating the status of executing the called intrinsic

Accept Free Phrase

Allows a free field format for low-volume data entry.

File Locking Capability

Permits a program to request exclusive access to an MPE XL or KSAM/V file through the usage of the EXCLUSIVE and UN-EXCLUSIVE statements.

Special Registers

TALLY, CURRENT-DATE, TIME-OF-DAY and WHEN-COMPILED are constructs frequently used programmatically and are provided by COBOL II/XL to reduce programming effort. These predefined registers eliminate the need for the user to construct such registers within the WORKING-STORAGE SECTION of the program.

Packed Decimal (COMP-3)

Allows for efficient storage of data during execution and results in performance efficiencies when using decimal arithmetic for computations.

Multiple Entry Points to Subprograms

Reduces code redundancy by allowing similar functions within identical portions of code to be coded into a single subprogram and be called through multiple names.

Data Management Interface

COBOL II/XL provides:

- Access to both sequential MPE XL and indexed sequential (KSAM/V) files through the usage of ANSI Standard COBOL Input and Output operations
- Access to Hewlett-Packard's screen management (VPLUS/V), data management (TurboIMAGE/V, ALLBASE) and graphics (DSG) products through the usage of procedure libraries
- Access from HPToolset which includes an editor, workspace manager, the ability to compile, link, and execute with debug and interactive symbolic debugging.
- Access to data definitions stored in Dictionary/V either standalone with COBOL II/XL or through HPToolset/XL allowing for the generation of COBOL Environment and Data Division entries from the dictionary automatically and concurrently with program development.
- The COBOL Definition Extractor utility is available to generate definitions for COBOL copylibs from System Dictionary/V definitions.

Compatibility

The COBOL II/XL compiler offers compatibility with the ANSI COBOL'74 Standard. Through the use of dual points of entry into the compiler, customers may choose which standard to execute, thereby, eliminating any need to convert between the two standards. The dual points of entry allow customers to avoid the existing incompatibilities between the COBOL'74 and COBOL'85 standards (e.g., reserved words), and thereby permit programmers to develop new code using the COBOL'85 standard while preserving their existing source code. Or if customers choose to move all of their applications to the COBOL'85 Standard, the compiler will flag the syntax changes allowing customers to modify, recompile, and execute using the ANSI COBOL'85 entry point.

System Environment

COBOL II/XL is supported on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product

Number	Product Description
31500A	COBOL II/XL Right-to-Use product
31500R	COBOL II/XL Right-to-Copy product with sublicense

Documentation

Part

Number	Document Name
31500-90001	HP COBOL II Reference Manual
31500-90002	HP COBOL II/XL Programmer's Guide
31500-90003	HP COBOL II Quick Reference Guide
31500-90004	HP COBOL II/XL Migration Guide
31500-90005	HP COBOL II/XL Reference Manual Supplement

Customer Training

Course

Number	Course Name
22832A	Self-Paced Training, Learning COBOL II

HEWLETT-PACKARD

Product Number 32116A

HP FORTRAN 77/V

For MPE V Based HP 3000 Systems

FORTRAN 77, the most recent ANSI FORTRAN standard, incorporates a number of improvements and extensions to FORTRAN 66. Among the more important are an IF-THEN-ELSE control structure, a CHARACTER data type and generalized Input/Output facilities. HP FORTRAN 77 is a superset of the ANSI FORTRAN 77 standard. It provides a well-defined language standard for software portability and offers extensions that enhance the flexibility and power of the FORTRAN 77 language.

HP FORTRAN 77/V is an implementation of FORTRAN 77 for the HP 3000 MPE V based computer systems. It is highly compatible with FORTRAN 66/V (formerly known as FORTRAN/3000).

HP FORTRAN 77/V provides:

- ANSI FORTRAN 77 (X3.9-1978) full language
- MIL-STD-1753 LANGUAGE EXTENSIONS
- HP language extensions
- MPE V system access
- FORTRAN 66/V compatibility support

Key HP FORTRAN 77/V Features

Program Structures

- SUBROUTINE and FUNCTION subprograms permit creation of modular programs.
- Multiple ENTRY points to a subprogram permit controlled sharing of data by different entry points.
- Structured programming statements - IF-THEN-ELSE, Block DO, and DO-WHILE — increase programmer productivity.

Data Manipulation

- INTEGER, REAL, and COMPLEX data types and operations provide efficient numeric computation.
- DOUBLE PRECISION and DOUBLE COMPLEX data types and operations provide extended precision numeric computation.
- CHARACTER data type and operations provide convenient management of strings.
- Bit manipulation functions provide shift, extract, and test operations.

Input/Output

- READ, WRITE, OPEN, CLOSE, and INQUIRE provide control and management of sequential or direct access formatted or unformatted files.
- List directed formatting, standard devices, ENCODE/DECODE, and NAMELIST provide simplified I/O programming.
- Internal files provide in-memory FORMAT conversion.
- Error handling is possible using the ERR=, END=, or IOSTAT=I/O specifiers in I/O statement.

Source Program Form

- INCLUDE provides insertion of program text (e.g. COMMON declarations) from another file.
- \$IF-\$THEN-\$ELSE directives provide conditional compilation.
- Names may have more than six characters to permit creation of meaningful program names.
- End of line comments provide convenient placement of program comments.

MPE System Access

HP FORTRAN 77/V provides easy access to MPE V system capabilities. This includes:

- Programmatic access to MPE V intrinsic functions
- Programmatic access to MPE V subsystems such as IMAGE/V, TurboIMAGE/V, VPLUS/V, Common Interface Intrinsics, and DSG
- Procedure call compatibility with other MPE V languages such as COBOL, FORTRAN 66/V, Pascal, and SPL

FORTRAN 66/V Compatibility Support

HP FORTRAN 77/V provides compatibility support to facilitate upgrade from FORTRAN 66/V.

- A Migration Guide describes FORTRAN 66/V to HP FORTRAN 77/V conversion issues.
- A migration aid, supplied with the compiler, translates FORTRAN 66/V features which have HP FORTRAN 77/V equivalents.
- Language extensions, provided by HP FORTRAN 77/V, support most FORTRAN 66/V features.

Data Specification

Data Type	Size (Bits)	Precision (Digits)	Magnitude Range
Integer	32	9	-2^{31} to $2^{31}-1$
Short Integer	16	4	-2^{15} to $2^{15}-1$
Real	32	7	-1×10^{-77} to 1×10^{77}
Double Precision	64	15	-1×10^{-77} to 1×10^{77}
Complex*	32	7	-1×10^{-77} to 1×10^{77}
Double Complex*	64	15	-1×10^{-77} to 1×10^{77}
Logical	32	—	.TRUE., .FALSE.
Short Logical	16	—	.TRUE., .FALSE.
Character	8	1 to 2^{31} characters**	ASCII Character Set (256 characters)

*A complex data item has a value of the indicated size and precision for both the real and the imaginary part.

**The string size is limited by available data space.

Compiler Features

- Local optimizations by the compiler include constant expression folding.
- Statement functions are implemented as macros.
- Dynamic array declaration and allocation in subprograms.
- Functions and subroutines called recursively.

Compiler directives provide:

- Selection of alternate semantics for DO loops, INTEGERS and LOGICALs, and data initialization.
- Control of program listing content.
- Interface with HP Toolset/V's symbolic debugger.
- Control of range checking, parameter checking and conformance to standard checking.
- Management of USL files and segments.
- Include files
- Conditioned compilation
- Up to 99 continuation lines

System Environment

HP FORTRAN 77/V is supported on MPE V based HP 3000 systems (T-MIT or later) and in Compatibility Mode on the 900 Series of HP 3000 systems.

A minimum of 1 Megabyte of memory is recommended.

Ordering Information

Software

Product Number	Product Description
32116A	HP FORTRAN 77/V Right-to-Use product
32116R	HP FORTRAN 77/V Right-to-Copy product with sublicense
31504A	FORTRAN 77 Developer's Package Right-to-Use product (Toolset/V & HP FORTRAN 77/V)
31504R	FORTRAN 77 Developer's Package Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
5957-4685	HP FORTRAN 77 Reference Manual
30000-90294	HP FORTRAN 77/V Reference Manual Supplement
5957-4686	HP FORTRAN 77 Programmer's Guide
31501-90005	HP FORTRAN 77/V Programmer's Guide Supplement
5957-4687	HP FORTRAN 77 Quick Reference Guide
5957-4690	FORTRAN 66/V to HP FORTRAN 77/V Migration Guide

Customer Training

Two training courses are available: a self-paced course with self study guide and videotape; a five-day classroom course

Course Number	Course Name
22961C	HP FORTRAN 77 Self-Paced Training
22959C	Programming in HP FORTRAN 77

HEWLETT-PACKARD

Product Number 31501A

HP FORTRAN 77/XL

For the 900 Series of HP 3000 Systems

FORTRAN 77, the most recent ANSI FORTRAN standard, incorporates a number of improvements and extensions to FORTRAN 66. Among the more important are an IF-THEN-ELSE control statement, a CHARACTER data type, and generalized Input/Output facilities. HP FORTRAN 77 is a superset of the ANSI FORTRAN 77 standard. It provides a well-defined language standard for software portability and offers extensions that enhance the flexibility and power of the FORTRAN 77 language.

HP FORTRAN 77/XL is an implementation of HP FORTRAN 77 for the 900 Series of HP 3000 systems. It provides access to subsystems and other languages on the 900 Series of HP 3000 systems. It is highly compatible with FORTRAN 66/V (formerly known as FORTRAN/3000), an implementation of FORTRAN 66.

HP FORTRAN 77/XL provides:

- ANSI FORTRAN 77 (X3.9-1978) full language
- MIL-STD-1753 language extensions
- HP language extensions
- MPE XL system access
- FORTRAN 66/V compatibility support

Key HP FORTRAN 77/XL Features

Program Structures

- SUBROUTINE and FUNCTION subprograms permit creation of modular programs.
- Multiple ENTRY points to a subprogram permit controlled sharing of data by different ENTRY points.
- Structured programming statements — IF-THEN-ELSE, Block DO, and DO-WHILE — increase programmer productivity.

Data Manipulation

- INTEGER, REAL, and COMPLEX data types and operations provide efficient numeric computation.
- DOUBLE PRECISION and DOUBLE COMPLEX data types and operations provide extended precision numeric computation.
- CHARACTER data types and operations provide convenient string management operations.
- Bit manipulation functions provide shift, extract, and test operations.

Input/Output

- READ, WRITE OPEN, CLOSE, and INQUIRE provide control and management of sequential or direct access, formatted or unformatted files.
- List directed formatting, standard devices, ENCODE/DECODE, and NAMELIST provide simplified I/O programming.
- Internal files provide in-memory FORMAT conversion.
- ERR=, END=, and IOSTAT=I/O specifiers allow I/O error handling.

Source Program Form

- INCLUDE statement allows insertion of program text (e.g. COMMON declarations) from another file.
- \$IF-\$THEN-\$ELSE directives provide conditional compilation.
- Names may have more than six characters to allow creation of meaningful program names.
- End-of-line comments provide convenient placement of program commentary.

Compiler Features

- The compiler permits specification of three levels of optimization:
 - No optimization (default) is necessary for symbolic debugging
 - Level 1 optimization, done over small code sections, produces smaller and faster code
 - Level 2 optimization, done over an entire procedure, produces the smallest and fastest code
- Statement functions are implemented as macros
- Dynamic array declaration and allocation in subprograms
- Functions and subroutines called recursively

Compiler directives provide:

- Selection of alternate semantics for DO loops, INTEGERS and LOGICALs, and data initialization
- Control of program listing content
- Interface to HP Toolset/XL's symbolic debugger and HP Symbolic Debugger/XL
- Control of range checking, parameter checking and conformance to standard checking
- Include files
- Conditioned compilation
- Up to 99 continuation lines

Data Specification

Data Type	Size (Bits)	Precision (Digits)	Magnitude Range
Integer	32	9	-2^{31} to $2^{31}-1$
Short Integer	16	4	-2^{15} to $2^{15}-1$
Real	32	7	-3.4×10^{-38} to 3.4×10^{38}
Double Precision	64	15	-1.7×10^{-308} to 1.7×10^{308}
Complex*	32	7	-3.4×10^{-38} to 3.4×10^{38}
Double Complex*	64	15	-1.7×10^{-308} to 1.7×10^{308}
Logical	32	—	.TRUE., .FALSE.
Short Logical	16	—	.TRUE., .FALSE.
Character	8	1 to 2^{31} characters**	ASCII Character Set (256 characters)

*A complex data item has a value of the indicated size and precision for both the real and the imaginary part.

**The string size is limited by available data space.

MPE System Access

HP FORTRAN 77/XL provides easy access to MPE/XL system capabilities. This includes:

- Programmatic access to MPE/XL intrinsics
- Programmatic access to MPE/XL subsystems such as TurboIMAGE/XL and VPLUS/V
- Procedure call compatibility with other MPE/XL languages such as COBOL II/XL, Pascal/XL and C/XL

Compatibility

HP FORTRAN 77/XL is completely compatible with HP FORTRAN 77/V, and highly compatible with FORTRAN 66/V.

- Migration documentation describes FORTRAN 66/V and HP FORTRAN 77/V to HP FORTRAN 77/XL migration issues.
- A conversion utility, supplied with the compiler, translates FORTRAN 66/V features which have HP FORTRAN 77/XL equivalents.
- Language extensions, provided by HP FORTRAN 77/XL, support most FORTRAN 66/V features which do not have HP FORTRAN 77/V equivalents.

System Environment

HP FORTRAN 77/XL is supported on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product Number	Product Description
31501A	HP FORTRAN 77/XL Right-to-Use product
31501R	HP FORTRAN 77/XL Right-to-Copy product with sublicense
31504A	FORTRAN 77 Developer's Package Right-to-Use product (Toolset/XL & HP FORTRAN 77/XL)
31508A	HP Symbolic Debugger/XL Right-to-Use product
31508R	HP Symbolic Debugger/XL Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
5957-4685	HP FORTRAN 77 Reference Manual
31501-90001	HP FORTRAN 77/XL Reference Manual Supplement
5957-4686	HP FORTRAN 77 Programmer's Guide
31501-90002	HP FORTRAN 77/XL Programmer's Guide Supplement
5957-4687	HP FORTRAN 77 Quick Reference Guide
31501-90004	HP FORTRAN 77/XL Migration Guide

Customer Training

Two training courses are available: a self-paced courses with self study guide and videotape; a five-day classroom course

Course Number	Course Name
22961C	HP FORTRAN 77 Self-Paced Training
22959C	Programming in FORTRAN 77

HEWLETT-PACKARD

Product Number 32102A

FORTRAN 66/V

For MPE V Based HP 3000 Systems

FORTRAN 66/V (formerly known as FORTRAN/3000) is a full implementation of the ANSI STANDARD FORTRAN (X3.9-1966). It also has many extensions which expand the capabilities and increase the power of the language.

Key FORTRAN 66/V Features

- Seven data types — integer, double integer, logical, real, double precision, complex, and character
- Character variables and character arrays
- Bit extract and deposit capability with partial-word designators
- Arrays with up to 255 dimensions
- Named common blocks initialized by block data subprograms
- Multiple entry points for subprograms
- Support of user-written error handling routines which are called under trap conditions
- Parameter statement for specifying constants with symbolic names
- Dynamic array declaration and allocation in subprograms
- Up to 99 files available during execution of a FORTRAN program
- Functions and subroutines called recursively
- Parameters to non-FORTRAN subprograms passed by value rather than reference
- ACCEPT and DISPLAY statements for free field input/output
- Compilation time editing
- Symbolic names with up to 15 characters
- Action labels specified in READ/WRITE statements to indicate point of transfer in case of end-of-file or I/O error
- Label used as an argument in subprogram call statements to indicate point of transfer in case of end-of-file or I/O error
- Mixed mode arithmetic
- Generic functions
- Built-in optional cross reference listing
- Undefined variable detection

Source Program Format

FORTRAN 66/V was designed with several powerful convenience features for interactive users. The nature of terminal devices makes the historical position-dependent fixed-format program representation inconvenient; however, FORTRAN overcomes these drawbacks by offering both fixed format and free format representation for source language input.

File Facility

Uniform access to disc files and standard input/output devices is accomplished through the MPE file system. You access your files using normal READ/WRITE statements.

Device type, file structure, and method of access can be defined at execution time; consequently, the devices used by a program can be readily changed. Access to KSAM files and IMAGE databases is supported by calls to the standard MPE intrinsics.

Sequential and random access of disc files is supported by FORTRAN 66/V.

If you have highly specialized requirements you may communicate directly with the MPE file system. Data file privacy is achieved through the normal MPE protection mechanisms.

System Environment

FORTRAN 66/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product Number	Product Description
32102A	FORTRAN 66/V Right-to-Use product
32102R	FORTRAN 66/V Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
30000-90040	FORTRAN 66/V Reference Manual
32102-90002	FORTRAN 66/V Pocket Guide

HEWLETT-PACKARD

Product Number 32106A

Pascal/V

For MPE V Based HP 3000 Systems

Pascal/V (formerly known as Pascal/3000) is Hewlett-Packard's implementation of the Pascal language for MPE V based HP 3000 systems. It provides several extensions to the ANSI/IEEE standards for Pascal. Pascal/V is particularly useful for development of large systems and subsystems.

Pascal offers a very rich and powerful set of data types and control flow structures. Along with control constructs, Pascal's basic building blocks of structure, namely procedures and functions, permit a top-down approach to program development. The modular self documenting characteristic of Pascal programs produces code which is easy to maintain and to enhance. As a further aid to program development, the language offers strong type checking and range checking capabilities. A significant portion of the debugging responsibility now rests with the compiler and the language.

Key Pascal Features

Program Structures

- Powerful control structures, such as IF-THEN-ELSE, CASE, WHILE-DO, and REPEAT-UNTIL, eliminate the need for GOTOs and provide structure to the programs, improving their readability.
- PROCEDURE and FUNCTION subprogram statements allow top-down, modular development of programs.

Data Manipulation

- REAL, LONGREAL, and INTEGER types provide efficient numeric computation.
- Character and string types, when coupled with pre-defined string handling functions, allow powerful string management operations.
- ARRAY, RECORD, SET, and FILE types facilitate the handling of data in a structured, user-defined format.
- Enumerated and subrange types provide further customization of standard data types.

Input/Output

- I/O routines support text (sequential only) and structured files (can be sequential or direct access).
- I/O functions including read, write, open, and close file functions, and other essential file management capabilities.
- Implicit data conversion for text files during input converts from ASCII to appropriate file variable format. The reverse is true for output.

Development Environment

- Integrated program development via HP Toolset provides symbolic debug, full screen editing, softkeys for compile and execute, and source and version management.
- Dictionary/V support for generation of TYPE and VAR declarations from previously stored definitions.

Language Standards

- Pascal/V is a superset of the ANSI/IEEE770X3.97-1983 standard for Pascal.
- Pascal/V conforms to the Hewlett-Packard standard for the Pascal language (excluding the module and conformist array features).
- The Hewlett-Packard Standard is also available on the HP 1000, HP 9000 and HP 3000 900 Series Computer Systems.

HP Standard Extensions to ANSI

The following are important extensions to ANSI/IEEE:

- STRING data type and associated functions
- Structured constants
- Constant expressions
- Direct access and other I/O extensions
- LONGREAL data type
- Structured function results
- Underscore in identifiers

Compiler Features

Compiler options facilitate control of:

- Source code, its conformance to ANSI or HP standards, and its conditional compilations
- Content of program listings
- Object code generated, and range checking
- Data layouts and their manipulation
- Separate compilation
- Interface to HP Toolset/V's symbolic debugger

Subsystem Access

- Provides access and interaction mechanisms for the MPE V file system, other MPE V languages (COBOL, FORTRAN, and SPL), and intrinsics.
- Provides access to data bases (HP SQL/V, IMAGE/V, or TurboIMAGE/V), VPLUS/V, Screen Management Intrinsics Library (Hi-Li), and other subsystems. Please see the relevant data sheets in this Specification Guide for more information.

System Environment

Pascal/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

A minimum of 1 Megabyte of memory is recommended. Pascal/V may use any number of extra data segments with a minimum of 7.

Ordering Information

Software

Product Number	Product Description
32106A	Pascal/V Right-to-Use product
32106R	Pascal/V Right-to-Copy product with sublicense
31505A	Pascal Developer's Package Right-to-Use product (Pascal/V & Toolset/V bundle)

Documentation

Part Number	Document Description
32106-90001	Pascal Reference Manual
32106-90002	Pascal Pocket Guide

HEWLETT-PACKARD

Product Number 31502A

HP Pascal/XL

For the 900 Series of HP 3000 Systems

HP Pascal/XL is Hewlett-Packard's implementation of the Pascal language for the 900 Series of HP 3000 systems. It allows several extensions to the ANSI/IEEE and ISO standards for Pascal, including those which provide it with system programming capabilities. These extensions open many new areas for Pascal usage. HP Pascal/XL will be particularly useful for development of large systems and subsystems, and those applications which require system level programming.

Pascal offers a very rich and powerful set of data types and control flow structures. Along with control constructs, Pascal's basic building blocks of structure, namely procedures and functions, permit a top-down approach to program development. The modular, self documenting character of Pascal's programs provide code which is easy to maintain and to enhance. As a further aid to program development, the language offers strong type checking and range checking capabilities. A significant portion of the debugging responsibility now rests with the compiler and the language.

HP Pascal/XL offers systems programming extensions to the above features. These extensions have permitted Pascal to be used extensively within Hewlett-Packard for systems level applications.

Key HP Pascal Features

Program Structures

- Powerful control structures, such as IF-THEN-ELSE, CASE, WHILE-DO, and REPEAT-UNTIL, eliminate the need for GOTOs and provide structure to the programs, improving their readability.
- PROCEDURE, FUNCTION and MODULE subprogram statements allow top-down, modular development of programs.

Data Manipulation

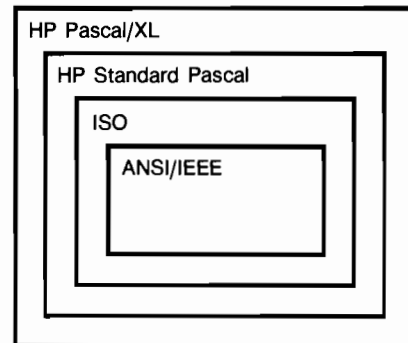
- REAL, LONGREAL, and INTEGER types provide efficient numeric computation.
- Character and string types, when coupled with pre-defined string handling functions, allow powerful string management operations.
- ARRAY, RECORD, SET, and FILE types facilitate the handling of data in a structured, user-defined format.
- Enumerated and subrange types provide further customization of standard data types.
- Conformant array parameter type provides flexibility in parameter passing.

Input/Output

- I/O routines support text (sequential only) and structured files (can be sequential or direct access).
- Convenient I/O functions including read, write, open, and close file functions, and other essential file management capabilities.
- Implicit data conversion for text files during input converts from ASCII to appropriate file variable format. The reverse is true for output.

Language Standards

HP Pascal/XL is a superset of the ANSI/IEEE770X3.97-1983 and ISO/DIS 7185 standards for Pascal. The systems programming extensions also make it a superset of the HP Standard for Pascal.



HP Standard Extensions to ANSI

The following are important extensions to ANSI/IEEE and ISO:

- STRING data type and associated functions
- Structured constants
- Constant expressions
- Direct access and other I/O extensions
- MODULES and separate compilation
- LONGREAL data type
- Structured function results
- Underscore in identifier

Systems Programming Extensions

HP Pascal/XL includes extensions which permit a very high degree of flexibility for data manipulation. Some of the salient features are:

- Control over Pascal's type checking
- Anyptr type and pointer manipulation routines
- Exception handling
- Procedure and Function variables
- Extended addressing
- Flexible procedure calling mechanisms

Compiler Features

Compiler options facilitate control of:

- source code, its conformance to ANSI or HP standards, and its conditional compilation
- content of program listings
- object code generated, and range checking
- data layouts and their manipulation
- intrinsic mechanism
- HP Toolset/XL's symbolic debugger and HP Symbolic Debugger/XL

The architecture of the 900 Series of HP 3000 systems permits special optimizing strategies. The following levels of optimizing are possible:

- No optimization (default), which is necessary during symbolic debugging
- Level 1 optimization, done over small code sections reduces code space and execution time
- Level 2 optimization, done over an entire procedure, results in the smallest and fastest code.

Subsystem Access

- Provides access and interaction mechanism for the MPE XL file system, and all other MPE XL languages and intrinsics. Users can also build their own intrinsics.
- Provides access to data bases (HP SQL and HP IMAGE), VPLUS, Screen Management Intrinsics Library (Hi-Li), and other subsystems. Please see the relevant data sheets in this Specification Guide for more information.

Compatibility

- HP Pascal/XL is a superset of the ANSI/IEEE770X3.97-1983, ISO/DIS 7185, and Hewlett-Packard standards for the Pascal language.
- HP Pascal/XL supports all features of Pascal as on MPE V based systems.
- High degree of compatibility with Pascal/V at the source code level. Please see the Pascal/XL Migration Guide for more details.
- A compiler option allows handling of data which is in MPE V format. It also aids in its conversion to MPE XL format.

System Environment

HP Pascal/XL is supported on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product Number	Product Description
31502A	HP Pascal/XL Right-to-Use product
31502R	HP Pascal/XL Right-to-Copy product with sublicense
31505A	Pascal Developer's Package Right-to-Use product (HP Pascal/XL & Toolset/XL bundle)
31508A	HP Symbolic Debugger/XL Right-to-Use product
31508R	HP Symbolic Debugger/XL Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
31502-90001	HP Pascal Reference Manual
31502-90002	HP Pascal Programmer's Guide
31502-90003	HP Pascal Quick Reference Guide
31502-90004	HP Pascal/XL Migration Guide

HEWLETT-PACKARD

Product Number 32115A

HP Business BASIC/V Interpreter and Compiler For MPE V Based HP 3000 Systems

Interpreter and Compiler

HP Business BASIC/V (formerly known as HP Business BASIC/3000) offers the complete language solution. With a variety of features to ease the programming task, HP Business BASIC supports the commercial applications programmer pursuing serious development as well as the novice programmer involved in ad hoc production of simple programs and tools. The interpretive environment provides immediate feedback on the effect of program modifications as well as indicating syntax errors as the program lines are entered. Extensive debugging tools also simplify application development. All or part of a program can be compiled resulting in significant performance improvements for most types of applications.

HP Business BASIC/V not only provides ease-of-use for the programmer, but also allows the programmer to write applications that are easy for the end-user. Run-time errors and interrupts from the keyboard can be trapped by the program and appropriate action taken. Statements are provided to support applications with softkey-based user interfaces.

Features

HP Business BASIC/V represents a significant feature set improvement over BASIC/V.

- Statements and constructs to support structured programming include: IF-THEN-ELSE, WHILE-ENDWHILE, REPEAT-UNTIL, SELECT-CASE, LOOP-ENDLOOP, FOR-NEXT, and GOSUB.
- All program development tasks are integrated. The interpreter and compiler, as well as all features of an *editor*, *debugger*, and *calculator*, are directly accessible through HP Business BASIC/V commands and statements.
- Callable subprograms and multi-line functions are supported with parameter passing and shared common data areas.
- Virtual program space allows large applications to be run when written in a modular fashion.
- The Program ANALYST, a code inspection tool, simplifies the task of optimizing, restructuring and maintaining applications.
- On-line HELP facility provides immediate information on syntax and functionality for all HP Business BASIC/V keywords and statements.
- Flexible Report Writer statements easily generate reports and tabular output.
- Advanced database statements provide easy but flexible data manipulation and access capabilities include retrieval, sorting, searching and automatic packing of buffers.

- Features, including user defined keys (softkeys) allow access to the display terminal.
- Intrinsics or dedicated HP Business BASIC/V statements interface to data entry forms (VPLUS/V).
- Run-time HP Business BASIC/V errors, database errors and end-of-file conditions can be detected and handled by the ON ERROR, ON DBERROR, and ON END statements.
- Full support for calling intrinsics for operating system services or access to other subsystems.
- HP Business BASIC/V programs can call external routines written in HP Business BASIC/V, PASCAL/V, SPL/V, COBOL II/V, and FORTRAN 77/V.
- Statements exist to issue MPE V commands and to run other programs from within a HP Business BASIC/V program.
- Long identifier names and alphanumeric labels improve program maintainability. Keywords can be used as identifiers, if desired.
- Seven data types and arrays up to six dimensions allow flexibility and efficiency in data structure design.
- MAT statements and functions simplify and improve the performance of array processing.
- CROSS REFERENCE (XREF) command easily generates a map of all definitions and uses of a program's identifiers.
- Extensive library of built-in functions for trigonometric, binary (logical), string and file operations.
- Conversion Utilities and tools, such as direct KSAM interface, JOINFORM (an HP 260 like forms package) and JOINFORM EDITOR (which creates and modifies JOINFORM files) are included for BASIC/V and BASIC/260 application conversion, respectively.
- Native language support is provided for date, time, and collating sequence.

Program Development Environment

Editing commands in HP Business BASIC/V include many found in dedicated text processors. The programmer can MOVE lines from one part of the program to another, or make a COPY of one or more lines. One can FIND all occurrences of a character string within the program and may CHANGE those occurrences to a different character string. Of course, there is a command for line editing (MODIFY) and a command to DELETE lines. As is traditional in BASIC implementations, a line can be replaced simply by entering the new line with the existing line number.

When listing a program, the LIST TO command will send the listing to a file or an output device. The INDENT command will "pretty-print" the program according to the programmer's specifications. Adding many lines of text is simplified by the AUTO command, which provides ascending line number prompts. Lines may be numbered up to 999999 and the RENumber command will resequence all or part of a program.

When a program is paused during execution (either by the PAUSE statement or by encountering an untrapped run-time error), variables and program lines may be displayed and modified. Execution can then be resumed with the CONTINUE command. Variables can be TRACED so that changes in their values are reported at run time. Lines can be TRACED so that their execution is reported. Programs can be run in STEP mode which displays each line as it is executed. The INFO, FILES, and CALLS commands give information on the current status of the interpreter environment.

Program ANALYST

The Program ANALYST is an environment within the Interpreter which displays information about the user's program in a softkey-driven, screen oriented easy to use format. The information displayed shows space utilization, date and time of last modification, software metrics (Halstead, McCabe etc.) and ordered statement frequency. The Program ANALYST can be used to analyze efficiency of data type usage, impact of compiler options, packing and unpacking of buffers, and much more. It can be used to assist in the process of dividing large program units into smaller ones. The combination of fast analysis of code and automated source-level transformations allows programmers to quickly restructure and tune existing applications.

Compiler

After programs have been developed and debugged using the Interpreter, performance can be significantly improved by using the Compiler. Commonly used routines can be compiled separately, stored in a Segmented Library (SL) and then CALLED from either interpreted or compiled programs. The Compiler can be invoked directly from the Interpreter with the COMPILE, COMPPREP, and COMPGO commands.

Integrated Application Tools

HP Business BASIC has been designed to provide complete functionality for the development of commercial applications. Database access, report generation, and screen management are integrated into the BASIC language.

Database Access

Statements are provided to interface to the TurboIMAGE/V database management system. All TurboIMAGE/V intrinsics have matching HP Business BASIC/V statements. In addition, HP Business BASIC/V provides search and sort capabilities. Data can be retrieved directly into program variables.

Report Writer

These statements simplify report generation tasks. Report Writer allows users to design their own report layouts, including page breaks, headers, trailers, page numbering, selective data presentation and totals tracking.

Forms Management

Rather than calling intrinsics to manipulate data entry forms, HP Business BASIC/V programmers can use simplified statements to display, write to, and read from forms. Data items are transferred directly between the screen and program variables.

Softkeys and Screen Management

Sets of softkey definitions can be stored in files and recalled with simple HP Business BASIC/V statements. When a program requests input from the user, special actions can be taken if the response is the pressing of a softkey. Cursor addressing and sensing, display enhancements, and the input of data from terminal memory are supported directly in the language.

Interface to External Software

HP Business BASIC/V provides the ability to call MPE V intrinsics and user-supplied routines written in HP Pascal/V and SPL/V. Other languages, such as COBOL/V and HP FORTRAN77/V, can be called with some restrictions on the types of the parameters. HP Business BASIC/V routines can be compiled and called from interpreted or compiled HP Business BASIC/V programs. When calling HP Business BASIC/V routines, features such as COMMON and error-trapping are available. HP Business BASIC/V routines cannot be called from other languages.

The SYSTEM statement allows a program to issue MPE V commands. The SYSTEMRUN statement provides a limited process handling capability. Other programs may be executed from the HP Business BASIC/V program or from within the HP Business BASIC/V Interpreter. The parent process can wait until the child process is finished or can proceed concurrently.

HP Business BASIC/V File System

A special file type is available (BDATA) which allows data to be stored with type information to facilitate retrieval later. In addition, there is full support for MPE V's ASCII and BINARY file formats. There are statements to CREATE and ASSIGN (open) files. When a file is open numerous built-in functions are available to return information on the file (record size, number of records, etc.) Various I/O statements provide great flexibility in managing the accessing of data files. One can PURGE, RENAME, COPY, or LOCK a file and POSITION its record pointer. For BDATA files, the I/O can be directed to a specific word within a record. The SEND OUTPUT statements will redirect the result of normal PRINT statements to files or devices.

BASIC/V to HP Business BASIC Conversion

HP Business BASIC/V includes tools to assist in the conversion of BASIC/V (formerly known as BASIC/3000) applications. Utilities are provided to automatically convert BASIC/V programs and data (BASD) files, as well as calls to IMAGE/V, TurboIMAGE/V, and BASIC/V KSAM routines. HP Business BASIC/V is highly compatible with BASIC/V on a statement-by-statement basis (i.e., 95-98%); however, differences in the organization of programs may result in the necessity for some manual modifications in order to make those applications run under the new system. HP Business BASIC/V allows for larger programs than BASIC/V though the maximum size of an individual program subunit is smaller. Large BASIC/V programs that need to be broken up into smaller subunits will benefit from use of the Interpreter code inspection tool, the Program ANALYST. After conversion, the Program ANALYST can be used to analyze the converted source code and suggest intelligent changes that will improve application performance.

BASIC/260 to HP Business BASIC/V Conversion

Note: Any reference to BASIC/260 pertains to both the BASIC/260 and BASIC/250.

The conversion of BASIC/260 applications involves the transfer of files to the HP 3000 (using data communications tools not included with the product) and conversion of those files on the HP 3000 itself. The conversion package will handle program files, data files, databases, and forms files. Though the compatibility between HP Business BASIC/V and BASIC/260 on a statement by statement basis is quite good, some differences in the behavior and functionality still exist. After the conversion package is run, a significant amount of manual conversion may be necessary, depending on programming style, due to environmental differences between the HP 260 and HP 3000 MPE V systems (eg. program unit size limitation differences). BASIC/260 conversions will also find the Interpreter code inspection tool, the Program ANALYST, to be helpful in breaking up large programs into smaller subunits. To improve performance of converted applications BASIC/260 programs should consider taking advantage of features designed explicitly for the HP 3000 (i.e., VPLUS forms instead of JOINFORMS). The Program ANALYST can also be used to analyze the converted source code and suggest intelligent changes that will improve application performance.

Data Type	Size (bits)	Range of Values
Short integer	16	– 32768 .. 32767
Integer	32	– 2,147,483,648 .. 2,147,483,647
Short real	32	– 1.157920E77 .. 1.157920E77
Real	64	– 1.15792089237161E77 .. 1.15792089237161E77
Short decimal	32	– 9.99999E63 .. 9.99999E63
Decimal	64	– 9.999999999999E99 .. 9.999999999999E99
String	8/char	length limited by data space available

Data Specification

HP Business BASIC/V allows data to be stored and manipulated in seven different representations, as shown at the bottom of this page.

System Environment

HP Business BASIC/V is supported on MPE V based HP 3000 systems (UB-Delta 1 MIT or later) and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

The right to use HP Business BASIC/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
32115A	HP Business BASIC/V Right-to-Use product
32115R	HP Business BASIC/V Right-to-Copy product with sublicense

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

Part Number	Document Description
32115-90001	HP Business BASIC/V Reference Manual
32115-90003	HP Business BASIC/V Programmer's Guide
32115-90004	HP BASIC/V to HP Business BASIC/V Conversion Guide
32115-90005	HP BASIC/260 to HP Business BASIC/V Conversion Guide

In addition, HP Business BASIC/V contains on-line documentation in the form of a HELP facility. This is equivalent to two-hundred pages of information on all HP Business BASIC/V statements and functions.



HEWLETT-PACKARD

Product Number 32111A

BASIC/V

For MPE V Based HP 3000 Systems

Interpreter and Compiler

BASIC/V (formerly known as BASIC/3000) is an easy-to-learn language designed especially for interactive terminal use. BASIC/V provides most of the features commonly found in a BASIC system. For a more advanced feature set, a follow-on product to BASIC/V is available. This product is HP Business BASIC/V, the commercial BASIC standard for MPE V and MPE XL HP 3000 computers. For more information see product number 32115A.

BASIC/V consists of both an interpreter and a compiler, which allows you to create and debug your BASIC programs interactively and then compile them for faster production. The BASIC/V Compiler provides the means for converting BASIC/V programs (including those which have been written, debugged, and saved via the BASIC/V Interpreter) into machine code. Compiled BASIC/V programs exist in the system as actual code segments and can be run directly, rather than through line-by-line interpreting.

Key BASIC Features

- Four numeric data types: real, integer, real extended precision, and complex.
- Mixed mode arithmetic.
- All standard functions (SIN, COS, LOG, etc.), plus matrices (up to 2 dimensions), strings, and files.
- Program segmentation with common storage.
- User definable security including password.
- Interpreter can be used alone or in conjunction with BASIC Compiler.
- Image database calls built into the language.
- Compiler supports all HP 3000 BASIC/V Interpreter language extensions.
- Compiler provides sharable machine code.
- Compiler can be combined with SPL/V procedures and FORTRAN/V subroutines into a BASIC/V program file.

Program Development

In the first phase, a BASIC/V program is usually written and debugged interactively using the BASIC/V Interpreter commands and statements. The interpreter constructs the interpretive version of a program. When you are satisfied that the BASIC/V program runs properly in its interpretive form, you save it (SAVE, FAST) in a file. This fast save file is the "source" input to the BASIC/V Compiler.

The BASIC Compiler is used to compile the fast save file. The program is then prepared in a form that results in an efficient machine code version of the original program.

The program executes directly under the operating system using the RUN command. BASIC/V programs may be compiled in batch or interactive mode, and programs may be run in either mode.

Character String Manipulation

The BASIC/V Interpreter incorporates the ability to define and manipulate ASCII character strings and string arrays. Substrings as small as zero characters and as large as 255 characters in length can be printed, concatenated, and compared to other strings.

Data Files

BASIC/V maintains three distinct file types:

- FORMATTED files provide advanced, easy-to-use capabilities that are intended for BASIC/V language use. These enable run-time checking of file data type.
- ASCII and BINARY files are available for communicating data to and from programs written in languages other than BASIC/V.

BASIC/V FORMATTED files may have a record size between 4 and 319 (16 bit) words. Data can be accessed either serially or on a record basis with random access to any record in the file. The ADVANCE and UPDATE statements provide the capability to access individual items within a record. Files may be created and purged either by commands or under program control.

Subroutines

BASIC/V provides two types of functions and two types of subroutines:

- Built-in functions include SIN, TAN, TNH (hyperbolic tangent). Approximately 40 such functions are provided.
- User-defined functions are established in the user's program and can be called from within the program. They may consist of multiple statements and local variables and arrays whose scope extends only within the declared function.
- A simple subroutine consists of a set of BASIC/V statements followed by a return statement. There is no explicit indication in a program as to which statements comprise a subroutine.
- External subroutines which are written in another language, e.g., FORTRAN 66/V or SPL/V, can be called by BASIC/V. BASIC/V programs may also call external subroutines from the libraries accessible to the user.

BASIC/V to HP Business BASIC/V Conversion

HP Business BASIC/V provides a conversion path for BASIC/V customers that want to move on to the 900 Series of HP 3000 computers. Utilities within HP Business BASIC/V automatically convert BASIC/V programs and data (BASD) files, as well as calls to IMAGE/V, TurboIMAGE/V, and BASIC/V KSAM routines. Although HP Business BASIC/V is highly compatible with BASIC/V on a statement by statement basis differences in the organizations of programs may result in the necessity for some manual modifications. In particular, large non-modular code segments will need to be broken up into the smaller subunit size of HP Business BASIC/V. A code inspection tool available in the interpreter of Business BASIC will assist in these and other post conversion modifications.

System Environment

BASIC/V is supported on MPE V based HP 3000 systems. BASIC/V is provided in Compatibility Mode on the 900 Series of HP 3000 systems in order to support existing BASIC/V applications.

Ordering Information

The right to use BASIC/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product

Number	Product Description
32111A	BASIC/V Right-to-Use product
32111R	BASIC/V Right-to-Copy product with sublicense

Documentation

Part

Number	Document Description
03000-90025	BASIC/V for Beginners
30000-90026	BASIC/V Interpreter Reference Manual
03000-90050	BASIC/V Pocket Guide
32103-90001	BASIC/V Compiler Reference Manual

HEWLETT-PACKARD

Product Number 32100A

SPL/V

For MPE V Based HP 3000 Systems

SPL/V is the Systems Programming Language for the MPE V based HP 3000 systems. Because it combines the efficiency of a machine-dependent language with the simple structure of a high-level language, SPL/V was used to write the operating system, language compilers, system utilities, and database management and data communications subsystems for the MPE V based HP 3000 systems.

Key SPL/V Features

- Self-documenting for ease of readability
- Permits access to all hardware features and data types
- Dynamic allocation of local storage for working space and local variables in procedures. Memory de-allocated on exit from procedures
- Program segmentation feature
- Assemble statement permits machine-level coding
- Six data types — logical, byte, integer, double integer, real, and long real

If you are used to developing your application software using assembly language, you will find that SPL provides the same efficient and powerful results while greatly reducing the development time. It allows you to write software quickly, easily, and efficiently, while producing object programs with good code compression and efficient execution times.

To simplify systems programming, SPL offers a high-level language similar to (but not equivalent to) ALGOL, together with features which enable you to easily exert control over machine-dependent functions of the computer system. You can operate directly on hardware registers, perform branches based on hardware status, extract/deposit/shift bit fields, or generate any sequence of hardware machine instructions. Also, commonly executed routines can be written in SPL and called from COBOL, BASIC, RPG, or FORTRAN programs.

The language provides many features normally found only in applications languages such as ALGOL and PL/1, and includes:

- Free-form structure
- Arithmetic and logical expressions
- High-level statements with unlimited nesting (IF, FOR, SWITCH, CASE, DO-UNTIL, WHILE-DO, MOVE, SCAN, assignment, and compound statements)
- Recursive procedures
- Variables and arrays of many data types

Variables

Variables may be either "global" or "local". Global variables are those declared in the main program and are accessible from any part of the program including procedures. Local variables, however, are declared within a procedure and are only accessible from within the procedure.

High-level Features

In all programming efforts, a need frequently arises for standard program constructs, such as loops, and evaluation of arithmetic expressions. Rather than hand-coding these often used structures each time, SPL allows you to write them at a high level. The compiler then provides an efficient, error-free code sequence in each case.

Machine-dependent Features

SPL allows the use of machine-level constructs to ensure complete control of the MPE V based systems. These constructs permit the following:

- Direct register reference
- Branching based on actual hardware conditions
- Bit extraction, deposit, and shift
- Generation of any sequence of hardware machine instructions (in the midst of high-level constructs)

System Environment

SPL/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

The right to use SPL/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
32100A	SPL/V Compiler Right-to-Use product
32100R	SPL/V Compiler Right-to-Copy product with sublicense

Documentation

Part Number	Document Description
32100-90001	Pocket Guide
30000-90024	Reference Manual
30000-90025	Textbook

Customer Training

Course Number	Course Name
22804A	HP 3000 SPL File System Introduction, 5 days HP Technical Center

HEWLETT-PACKARD

Product Number 32104A

RPG/V

For MPE V Based HP 3000 Systems

RPG, or Report Program Generator, is a machine independent, problem-oriented report generating language that is easy to learn, use, and code. It allows you to specify many important operations with a minimum of effort by making simple entries on specially formatted coding sheets. HP's RPG is highly compatible with the RPG II languages offered by other vendors. Minimal effort is required to upgrade standard RPG II programs from other vendor's systems to HP. In addition, the RPG compiler helps detect errors at the source language level with extensive diagnostic messages.

Key RPG Features

- Edit codes
- Calculation control of I/O
- Cross reference
- Formatted dump
- Preselected or dynamic run time error options
- Source level debugging with DEBUG operator
- Data structures
- LDA (Local Data Area); UDS (User Data Structure)
- Command key and Function key indicators
- Message catalog access with MSG and DSPLM
- Figurative literals *BLANK(S), *ZERO(S)
- Short-hand arithmetic
- Relative end positions
- In-program array sorting with SORTA
- Buffer integrity checking
- Structured programming constructs

RPG can handle jobs ranging from simple tasks such as printing address labels to very complex ones such as an entire payroll process (printing paychecks, payroll registers, and various allied reports). RPG is ideal for producing such documents as inventory lists, billings, invoices, insurance benefit notices or summaries of sales, profits and losses, and customer transactions. It is also excellent for updating the master files used in producing these documents.

Extensions to RPG II

Parameters for external subroutine calls: Parameters may be specified after an EXIT (external subroutine call) operation, simplifying interfacing with COBOL, SPL, BASIC, or FORTRAN subroutines.

Interface to database management: Databases can be accessed through regular I/O reads and writes by specifying the file as being an IMAGE database in the file specification section. Keyed sequential access method (KSAM) files with up to 16 keys may also be accessed using regular I/O operations.

Run-time error options: Three methods are provided for handling run-time errors:

1. Specifying on the Control Record at compile time whether the run-time error should be ignored or the program terminated.
2. Allowing the operator to determine the mode of operation at run-time.
3. Testing an error code in RPG calculations and determining the mode of operation programmatically.

Cross reference option: A cross reference may be requested showing all references to file names, indicators, and field names.

Automatic program segmentation: RPG will automatically segment code generated for an RPG program in programmer-selectable 2K, 4K, 6K or 8K-byte segments.

EDBDIC/ASCII automatic translation: You can request RPG to automatically generate file translation tables for EBCDIC to ASCII or ASCII to EBCDIC conversions, or to use an alternate collating sequence.

Interface to terminal screen management: The interface between RPG and VPLUS/V allows you display forms on HP block-mode terminals, flag input data fields which are in error, and simplify other data entry functions. You also have the ability to do function key labeling.

An alternative interface, the RPG Screen Interface, allows you to perform screen access operations similar to "SDA" screens on other vendors' systems.

Calculation indicator repetition: Duplicate conditioning indicators need not be repeated line-to-line in calculations.

Buffer integrity checking allows you to ensure data integrity for KSAM and MPE files when accessing one of those files more than once from within the same program.

Data Types

RPG allows data to be input or output in the following formats:

- Binary — one or two word binary data
- Packed decimal
- Alphanumeric
- Unpacked decimal
- Unpacked decimal with leading or trailing sign

Specification Types

The statements that describe the input, processing, and output to the compiler must be written according to the rules of RPG. The eight specification types are:

- Header (Control record)
- File description
- Extension
- Line counter
- Input
- Calculation
- Output
- Array/Table data

File Sort Utility

Extra Function Sort for RPG/V (XSORT) provides all the capabilities of SORT/V, plus the following:

- Selection by "Include" and "Omit" specifications of a subset of the input records prior to sorting
- Reformatting of data records on output after sorting

RPG Interactive System Environment

The RPG Interactive System Environment (RISE) is a specialized editor dedicated to the creation and maintenance of RPG programs. It presents a friendly user interface to a development environment in which a user can easily create and edit source code, as well as compile, prepare, and run the program.

Features

- Visual editing with images that represent the RPG Specification Record forms
- Command menu and special function keys (softkeys) for a broad range of uses
- Page-at-a-time direct screen editing
- Ability to call the RPG compiler as well as the Segmenter and manipulate the compiled listing using a split screen
- Choice of editing a file directly or editing a copy of that file
- Recoverability feature to restore deleted lines
- RISETOUR, an automatic instructive guided tour to quickly acquaint a new user with the system

RISE runs on Hewlett-Packard terminals which support block mode and have at least 8K bytes of memory and the display enhancement set.

RPG Screen Interface

The RPG Screen Interface (RSI) provides simple, easily understood commands for workstation screen access.

Features

- Compatible with other vendors' "SDA" screens
- Indicators set in RPG program can dynamically alter field attributes on screen
- CONSOLE files automatically create forms from Input specifications
- Forms are compiled into library file using the utility program SIGEDIT.PUB.SYS (RPG Screen Interface Editor)
- "Command Keys" implemented for compatibility with other vendors' terminals

RPG Copylib Preprocessor

The RPG Copylib Preprocessor (RPGCOPY) is an optional preprocessor utility which can be used to include RPG source records from other files into the source code being compiled.

RPG Initializer Utility

The utility program RPGINIT.PUB.SYS creates and initializes two temporary, special-purpose files:

- LDAFILE — contains up to 32, 256-byte records initialized to blanks. This file is useful for programs specifying access to the Local Data Area (LDA). The default size is 2 records.
- RPGUPDATE — contains a single 80-byte record initialized to the system date in the form MMDDYY. This file is useful for programs specifying UPDATE initialization to be from a file.

System Environment

RPG/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on the 900 Series of HP 3000 systems.

Ordering Information

Software

Product Number	Product Description
32104A	RPG/V Compiler
32104R	RPG/V Compiler Right-to-Copy product with sublicense.

Documentation

Part Number	Document Description
32104-90001	RPG Reference Manual
32104-90006	RPG Utilities Reference Manual
32104-90003	RPG Listing Analyzer (Info on RPG options and specification templates)

HEWLETT-PACKARD

Product Number 31506A

HP C/XL

For the 900 Series of HP 3000 Systems

A Highly Portable Programming Language

HP C/XL is Hewlett-Packard's implementation of C for the 900 Series of HP 3000 systems. It is a general-purpose programming language that features modern control flow and data structures, a rich set of operators, and economy of expression. It combines the convenience and portability of a high-level language with the flexibility and efficiency of assembly language. While C is the systems programming language for UNIX(1), it is also widely used for applications programming. And due to its portable nature, C is regarded as an excellent vehicle for transporting software between systems.

Key Features

Program Structures

- Functions are recursive and can return scalar structure, or union values, or `void`.
- Block structured scope of variables is provided.
- Public and private functions and data is provided.
- Structured programming statements; `if-else`, `for`, `while`, `switch`, are available.
- Pragmas are provided to specify optimization options, copyright information, version identification, listing control, and MPE XL intrinsic access.
- Access to MPE XL run command arguments `PARM` and `INFO` via `argc` and `argv`.

Data Manipulation

- `short`, `int`, `long`, `float`, `double`, and `char` provide integer, floating point, and character data types and operations.
- `short`, `long`, `signed`, and `unsigned` provide control of size and interpretation of numeric data items.
- `auto`, `static`, `extern`, and `register` permit control of allocation of data items.
- Arrays, structures, and unions provide composite data types.
- Pointers provide effective manipulation of addresses.

Input/Output

- `scanf`, `printf`, `fopen`, `fclose`, and their variants provide control and management of sequential, formatted files.
- Standard input and output files simplify programming.
- Standard error file is available for error messages.
- In-memory format conversion is available.
- Library functions provide character and string handling, as well as dynamic memory allocation.

Source Program Form

A macro preprocessor provides a variety of source program management functions.

- `#define` statement allows definition of macros.
- `#include` statement allows inclusion of program text (e.g. common declarations) from another file.
- `#if-#else-#endif` directives provide conditional compilation.
- Identifiers may have more than eight characters to allow creation of more meaningful names.

Compiler Features

- The compiler permits specification of three levels of optimization:
 1. No optimization (default); this level is required for symbolic debugging.
 2. Level 1 optimization produces smaller and faster code.
 3. Level 2 optimization produces the smallest and fastest code.
- HP C/XL is supported by HP Symbolic Debug/XL.
- HP C/XL provides a compiler listing with tables that report code and data offsets.

Subsystem Access

- Provides access to MPE XL file system, intrinsics and all other MPE XL languages.
- Provides access to databases (via TurboIMAGE and HPIMAGE), VPLUS, High Level Screen Management Intrinsic Library (Hi-Li) and other subsystems.

Compatibility

HP C/XL is completely compatible with HP C/HP-UX for the HP 9000 Series 800 and is highly compatible with C/9000, VAX C(2). In addition, HP C/XL is highly compatible with CCS/C 3000 (for MPE V based systems) and CCS/C 1000 both from Corporate Computer Systems(3).

- Documentation included with the product describes system compatibility guidelines.

System Environment

HP C/XL is supported on MPE XL based HP 3000 systems.

Ordering Information

Software

Product

Number	Description
31506A	HP C/XL Compiler Right-to-Use product
31506R	HP C/XL Compiler Right-to-Copy product with sublicense
31507A	HP C/XL Developer's Package Right-to-Use product (HP C/XL Compiler and HP Symbolic Debugger/XL)
31508A	HP Symbolic Debugger/XL Right-to-Use product
31508R	HP Symbolic Debugger/XL Right-to-Copy product

Documentation

For further technical information, consult the following documents:

Part

Number	Document Description
92434-90001	HP C Reference Manual
92434-90002	HP C Programmer's Guide
92434-90003	HP C Quick Reference Guide
31506-90001	HP C/XL Reference Manual Supplement
30026-90001	HP C/XL Library Reference Manual
92435-90001	HP Symbolic Debugger User's Guide
92435-90002	HP Symbolic Debugger Quick Reference Guide

(1) UNIX is a trademark of AT&T.

(2) VAX is a trademark of DEC. "VAX C" refers to C running under UNIX 4.2 BSD.

(3) CCS/C compilers are marketed and supported by Corporate Computer Systems, Inc., 33 West Main Street, Holmdel, New Jersey 07733 Tel. (201) 946-3800.

HEWLETT-PACKARD

Product Number 31508A

HP Symbolic Debugger/XL

For the 900 Series of HP 3000 Systems

A Powerful Symbolic Debugger

HP Symbolic Debugger/XL is a powerful, full-featured symbolic debugger to help the user locate and correct errors in programs. The feature set facilitates high programmer productivity and user friendliness. HP Symbolic Debugger/XL is the interactive source level debugger for the HP C/XL, HP FORTRAN 77/XL and HP Pascal/XL languages and runs under the MPE XL operating system on the 900 Series of HP 3000 systems.

HP Symbolic Debugger/XL allows the user to examine the program state in which an error or some other condition occurs. The user may then take corrective action and resume execution, or abort the program.

Key Features

- Provides language sensitive expression evaluation for HP C/XL, HP FORTRAN 77/XL and HP Pascal/XL.
- Friendly user interface including online help, support of multiple skill levels, and windowing-orientation that supports up to three panes, allowing viewing of source statements and assembly instructions simultaneously.
- Breakpoints can be set, deleted, suspended and activated at specified locations allowing the user to examine the state of a program.
- Single-stepping through a program or over procedures is provided at the source statement or the equivalent assembly instruction level.
- Assertions allow the execution of a list of commands before every source statement for tracing elusive bugs.
- Allows the display and modification of program variables.
- Multi-level stack tracing allows the user to trace program flow.
- Extensive interrogation of the debugger state, including breakpoints, files, globals, procedures and their variables is provided.
- Record and playback of debugger commands to help recreate debugging sessions.
- Provides access (within the debugger) to the MPE XL operating system via the command interpreter.
- Macro facility allows users to define their own names for command strings.

Functional Description

Source level symbolic debugging: While in the debug process, the HP Symbolic Debugger/XL recognizes the names, types and locations of all the variables and routines for HP C/XL, HP FORTRAN 77/XL and HP Pascal/XL. Programs can be debugged without the need for load maps, symbol table dumps or mixed listings.

Assembly level debugging: The debugger allows the user to display high-level source code in assembly language. It can single-step at the instruction level and supports display and modification of registers.

Interactive debugging process: The user interacts with the program as it executes and can examine or alter variable values of the running program without having to insert statements into the code.

Separate Process: HP Symbolic Debugger/XL runs as a process separate from the user program. No significant code or data space is lost and no extra statements are added in order to debug. The program being debugged runs exactly the same as it would normally. There is no need to restructure the program to debug it.

Program breakpoints: HP Symbolic Debugger/XL supports user set breakpoints to monitor program execution, for example, to halt if a variable reaches a specified value. Any number of possible paths can be trapped and values can be quickly tracked through the program to determine error conditions.

Supports command files and message logging: Debugging sessions may occur where users submit debug commands in a file, and have results logged in another file. This automates the debugging process, so users don't have to wait for bugs whose symptoms may take a prolonged period of time to occur. In addition, the user can use commands recorded from a previous debugging session and edit the played back version (if desired) for use in the current session.

Compatibility

Detailed compatibility information is contained in the HP Symbolic Debugger User's Guide.

- HP Symbolic Debugger/XL is compatible with the HP Symbolic Debugger/HP-UX product, which runs on the HP 9000 Series 800 product lines.

Terminal Considerations

- The window-oriented feature is implemented through the use of memory-locking, therefore, only terminals that support memory-locking will support this feature. Terminals that do not support memory-locking may be used with the product but will not support the window feature.

System Environment

HP Symbolic Debugger/XL is supported with HP C/XL, HP FORTRAN 77/XL and HP Pascal/XL on the 900 Series of HP 3000 systems.

Order Information

Software

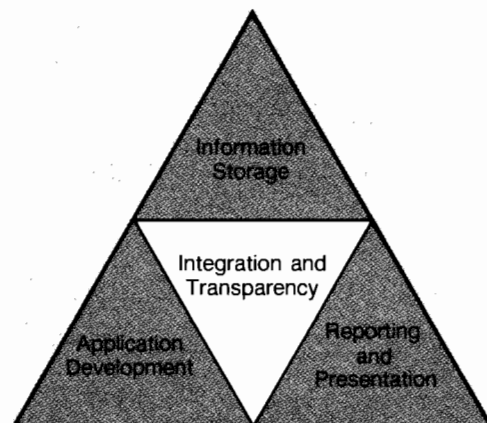
Product Number	Description
31508A	HP Symbolic Debugger/XL Right-to-Use product
31508R	HP Symbolic Debugger/XL Right-to-Copy product with sublicense

Documentation

For further technical information, consult the following documents:

Part Number	Document Description
92435-90001	HP Symbolic Debugger User's Guide
92435-90002	HP Symbolic Debugger Quick Reference Guide

Integration and Transparency



HEWLETT-PACKARD

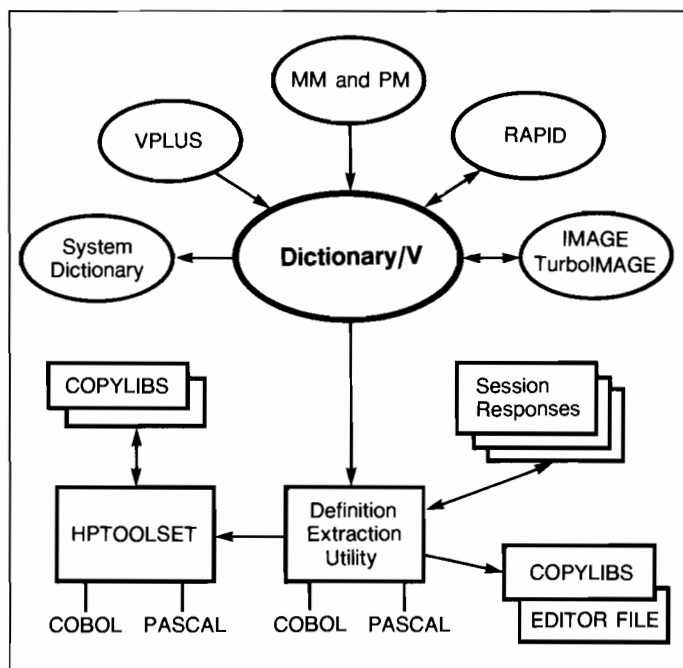
Product Number 32244A

HP Dictionary/V For MPE V Based HP 3000 Systems

HP Dictionary/V (formerly HP Dictionary/3000) is a data dictionary and data directory facility that provides the means to control and coordinate an organization's data processing resources more efficiently. The data dictionary consists of an TurboIMAGE database which stores information about the data processing environment, a high-level user interface facility, and a set of powerful utilities. In the interactive mode, the dictionary interface accepts commands and then generates prompts to guide the user through entry, update, deletion, and reporting operations. Dictionary/V's friendly dialogue makes it easy to use and maintain.

System Dictionary is HP's standard data dictionary on the HP 3000. While integration of subsystems with System Dictionary is in progress, customers requiring this integration should purchase Dictionary/V. Customers using Report and Inform, or requiring links to Materials Management and Production Management should purchase Dictionary/V.

To support Dictionary/V customers who intend to eventually convert to System Dictionary, a Dictionary/V conversion utility is included with System Dictionary.



Dictionary/V integration with HP subsystems

Key HP Dictionary/V Features

- **Centralized Data Definitions**
Provides one consistent source for all data definitions and locations. Documents TurboIMAGE, KSAM, MPE and VPLUS forms files. Documents programs and TurboIMAGE security.
- **Program Development**
Supplies data definitions for COBOL, Pascal and Transact. Provides flexible data environment documentation through built-in reporting capabilities.
- **Report Generation**
Supplies definitions for Report and Inform. Supports relational user views of the data for ad hoc reporting with Inform.
- **Ease of Use**
Loads TurboIMAGE database definitions into the dictionary automatically. Loads definitions for VPLUS forms files into the dictionary automatically. Provides interactive user commands and prompts for dictionary entry.
- **Database Design and Maintenance**
Generates TurboIMAGE schemas and root files automatically. Facilitates unloading and loading TurboIMAGE databases to tape or disc — including edits to assist in restructuring. Reports structural and usage statistics for maintenance decisions.

■ Links to other Subsystems

Links to Materials Management and Production Management. The customizer will automatically update Dictionary/V to reflect changes to those subsystems.

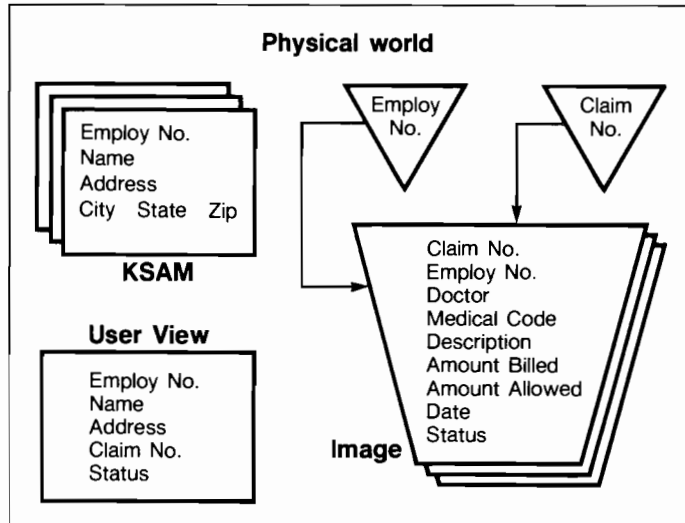
A Programmer's Tool

Dictionary/V is a guide to what data is available; a tool for monitoring and managing change; and an aid to program design. **New applications and modifications to existing ones can be completed faster because there is one source which can resolve many of the designer's questions** such as: What data is available, where is it, who is responsible for it, and what does it look like? The impact of system changes can be quickly analyzed since the files, programs, and users that are involved can be identified. **The structure of systems and databases can also be evaluated because hierarchical relationships are stored in the dictionary.**

Inconsistencies between data element definitions become a problem of the past when the dictionary is designed into the system from the beginning. Designers can question the dictionary about the standard attributes of the elements before any coding is done. Duplicated effort is minimized and time is saved during both development and maintenance.

Dictionary/V provides information for program development. Data definitions are provided automatically for Transact and Report programs. Both data definitions and locations are provided for Inform programs. File and element definitions for inclusion in both environment and data divisions are provided by the COBOL extract utility. Type and Var declarations for inclusion in Pascal programs are provided by the Pascal extract utility. **When data definitions all come from one consistent source, program quality improves and debugging problems are reduced.**

Dictionary/V provides logical views of the data for Inform users. Specifications for logical views are stored in groups within the dictionary. These groups can include fields from different files (TurboIMAGE, KSAM, and MPE), and different TurboIMAGE databases. The logical groupings are explicitly used by Inform to provide a friendly way to access data. The dictionary isolates the logical view from the physical structure of the data processing environment. Dictionary/V and Inform work together to provide a simple inquiry and reporting facility for both the end-user and the programmer. **With Inform, programmers spend less time on report generation.**



Relational User Views

A DBA's Tool

Dictionary/V is a repository for data definitions. This helps ensure consistency of data items and is a key for setting standards. The extensive documentation facility within the dictionary provides for documenting not only the data, the files, and the programs, but also the relationships between them. A user definable entity is also provided for customization of the dictionary. This definable entity is called a Category.

```

>show file
FILE ORDMGT
TYPE: RESPONSIBILITY:
BASE
FILE(ALIAS): TYPE: MAST FILE(PRIMARY): CAPACITY:
PRODUCT ELEMENT(ALIAS): PROD-NO PROPERTIES: ELEMENT(PRIMARY):
DESCRIPTION * U (8,0,8) PROD-NO
X (30,0,30) DESCRIPTION
FILE(ALIAS): TYPE: DETL FILE(PRIMARY): CAPACITY:
SALES ACCOUNT * X (10,0,10) ACCOUNT
CHAIN MASTER SET: CUSTOMER
SORTED BY: PURCH-NO
PROD-NO * U (8,0,8) PRDD-NO
CHAIN MASTER SET: IPRODUCT
QUANTITY P (7,0,4) QUANTITY
PRICE P (11,0,6) PRICE
TOTAL P (11,0,6) TOTAL
PURCH-DATE * X (6,0,6) PURCH-DATE
CHAIN MASTER SET: DATE-MASTER
DELIV-DATE * X (6,0,6) DELIV-DATE
CHAIN MASTER SET: DATE-MASTER
PURCH-NO U (6,0,6) PURCH-NO

```

Dictionary "Output"

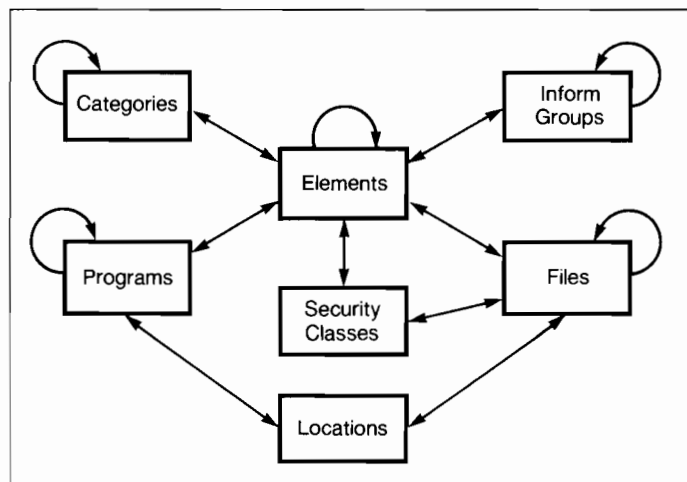
Dictionary/V simplifies the creation and testing of new databases by allowing designers to leverage off of existing data definitions, and providing the flexibility to set up different database designs and view them with the Dictionary/V reporting commands. System designers can therefore concentrate on the design instead of the coding of new database schemas. The database creation utility will create a root file and schema automatically from the complete design. **Database maintenance is also made easier** by the selective load and unload feature of the database utilities.

Dictionary Contents

The dictionary contains entries which are meaningful to the data processing and end-user environments. The gap between the physical data layout and the user's relational view of the data can now be bridged through the dictionary.

The dictionary captures a detailed description of the physical information system. This blueprint qualifies and quantifies all the entities in the system. These include:

Elements	Locations	Inform Groups
Files	Security Classes	
Procedures	Categories	



Dictionary/V Organization

Dictionary "Input"

Dictionary/V can be easily integrated into an existing data processing environment or can be designed into the information system from the start. The Database Definition utility automatically loads the data element definitions and structure of an existing TurboIMAGE database into the data dictionary. The VPLUS Forms Definition utility automatically loads the definitions of existing forms into the dictionary. When new programs, files, and elements are added to the system, the dictionary needs to be updated to reflect the changes. A combination of user-issued commands and system-generated prompts allows the dictionary to be efficiently updated.

```
> create file
      FILE credit
      LONG NAME Customer Credit
      TYPE Detl
      RESPONSIBILITY Finance Manager
      DESCRIPTION Credit includes CUSTNO, CPLMIT, and CPBALANCE.
      This is updated on a per order bases.
```

Dictionary "Input"

Dictionary "Output"

There are several ways to access the valuable information which has been stored in the dictionary. Reports can be generated for reference and documentation purposes. Four flexible output commands support interactive inquiry of the dictionary contents. Now programs can be documented quickly and accurately. This not only saves time during the development cycle but also during maintenance operations. Standardized reports include element, file, and relationship listings which can be selected and sorted to meet specific user criteria.

```
> display element
      ELEMENT custno
      ELEMNT:      TYPE: SIZE: DEC: LENGTH: COUNT: RESPONSIBILITY:
      CUSTNO      9      10      0      10      1      Order Processing
      LONG NAME: CUSTOMER NUMBER
      HEADING TEXT: CUSTNO
      ENTRY TEXT: Please enter customer number:
      EDIT MASK:
      MEASUREMENT UNITS:
      BLANK WHEN ZERO: NO
      DATE CREATED: 81/07/14 BY MGR
      DATE CHANGED: 81/07/14 BY MGR
      DESCRIPTION:
      Unique number which is assigned at order time.
      Divisions of a major account do not get their own
      number. Use the ship to address to differentiate.
```

Dictionary "Output"

Dictionary Utilities

- Database Create
- Database Definition
- VPLUS Forms Definition
- COBOL Definition Extraction
- Pascal Definition Extraction
- Database Unload
- Database Load
- Database Audit

Database and VPLUS Forms Integration

There are two utilities which load information into the dictionary and one which helps create an TurboIMAGE database from dictionary information. The Database Creation utility uses the information stored in the dictionary to automatically produce a root file for a given database. Since information about the database is collected in an interactive exchange between the dictionary and the designer, the chance for mistakes is reduced.

The Database Definition Utility can be used to load the definition of an existing database into the dictionary. This automatic procedure saves entering the schema definition into the dictionary manually. It also gives the TurboIMAGE designer the flexibility to either create a database first or create its dictionary definitions first.

The VPLUS Forms Definition utility creates entries in the dictionary for forms files, forms, and data fields. The associations of the data fields to the forms and the relationship between the forms and the forms files are performed automatically. Duplication and incompatibilities between new and existing forms are identified and easily corrected.

Program Development

There are two extract utilities that facilitate program development. The COBOL Definition Extraction utility generates file and element definitions from the dictionary for both environment and data divisions. Code can be generated for any type of file (TurboIMAGE,KSAM, MPE or VPLUS) or for a single element in the dictionary. Standard and special parameters (the DB-status array, modes and lists, base password, and search item parameters) can also be generated. The COBOL utility allows for the definition of arrays, implicit and explicit redefines, and the VPLUS comarea. The ability to define prefixes, use alias names and change options affecting the generated code, make this code customizable for matching local COBOL coding standards. Session responses are logged to an editor file. This file could then become the foundation of a job stream for invoking the utility.

The Pascal Definition Extraction utility generates Pascal Type and Var declarations for inclusion in a Pascal program. Similar to the COBOL utility, definitions can be generated for any type of file or for a single data element in the dictionary. The utility allows for the definition of arrays, element back references, standard TurboIMAGE parameters and the VPLUS comarea. Session responses are also stored in a log file for subsequent inclusion in a job stream.

The generate utilities can be used in either of two ways. First, the user can run them stand-alone. In this mode COBOL definitions go into a copylib and Pascal definitions go into an editor file. Alternatively, the utilities can be called from inside the HPToolset environment. In this mode the generated definitions go directly into the file or copylib being edited. The copylib capabilities of the COBOL utility allow programmers to create their copylibs automatically and concurrently with program development instead of waiting until some later date when resources are available.

Maintenance

The dictionary utilities support the restructuring of TurboIMAGE databases. By unloading the database, making changes to the dictionary, creating a new TurboIMAGE root file and schema, and loading the existing contents into the new database, structure changes can be made. Possible changes include adding or deleting data sets or items, re-arranging data sets or items within a data set, or changing an item's attributes. Because a database can be unloaded to a disc file, the unload process takes less time than it would with TurboIMAGE utilities. Since the disc file does not have to be predefined, the DBA does not have to determine how large a file to build.

The dictionary makes it possible to plan maintenance activity more efficiently. The Database Audit utility provides statistics on synonyms in both master and detail sets, chain lengths, and integrity of the chains. If it is determined that a database needs to be reloaded to increase performance or to fix a chain linkage problem, the Database Load and Unload utilities speed up this process by supporting disc backup files and selective database loading and unloading. Maintenance efforts can be reduced by using the dictionary as a central reference point to determine the impact of changes. The documentation of descriptions, users, and locations also increases the productivity of maintenance programmers.

Dictionary Security

Five levels of access exist to the data held in the dictionary. At the time the dictionary is created, the DBA has the option to assign a password to each of the five levels. These levels control a full spectrum of security within the dictionary — from total access to read only access. These levels may be redefined at any time the DBA feels it is necessary. The extensive security system allows the DBA to maintain control and to ensure the dictionary's integrity.

Dictionary/V can be used to document the security that has been assigned to the different files defined in the dictionary. In the case of TurboIMAGE files, this security is used when the TurboIMAGE root file is created by the Dictionary/V utility.

Dictionary/V also provides security for Inform that prevents unauthorized persons from gaining read access to the dictionary by limiting the names of the database files and Inform groups displayed on the Inform menus. When Inform security is set up in the dictionary, Inform menus will display only the information granted a user via an Inform password.

System Environment

Dictionary/V is supported on MPE V based HP 3000 systems and in Compatibility Mode on MPE XL based HP 3000 systems.

Ordering Information

The right to use Dictionary/V software on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

Product Number	Product Description
32244A	HP Dictionary/V Right-to-Use product.
32244R	HP Dictionary/V Right-to-Copy product with sublicense.

For customers who want to use Dictionary/V in combination with Transact, Report, Inform, HPToolset, COBOL, or Pascal, several packages are available. Please contact your local sales office for details.

Documentation

Part Number	Document Description
32244-90001	HP Dictionary/V Reference Manual

Customer Training

A self-paced training course is available. The course includes instruction and labs. Training is provided on:

- Creating and updating Dictionary/V.
- Documenting TurboIMAGE, MPE, KSAM, and VPLUS files.
- Extracting definitions for COBOL and Pascal programs.
- Creating Inform user groups.
- Using the utilities that maintain and create TurboIMAGE databases.

A classroom training course is also available.

Contact the local sales office for a course outline and schedule.

Course Number	Course Name
22843B	Using Dictionary/V (HP Dictionary/V Self-paced Training)
35013B	Dictionary/Inform/Report Classroom Training (3 Days)

HEWLETT-PACKARD

Product Numbers 32254A, 32255A, 32256A, and 32257A

HP System Dictionary

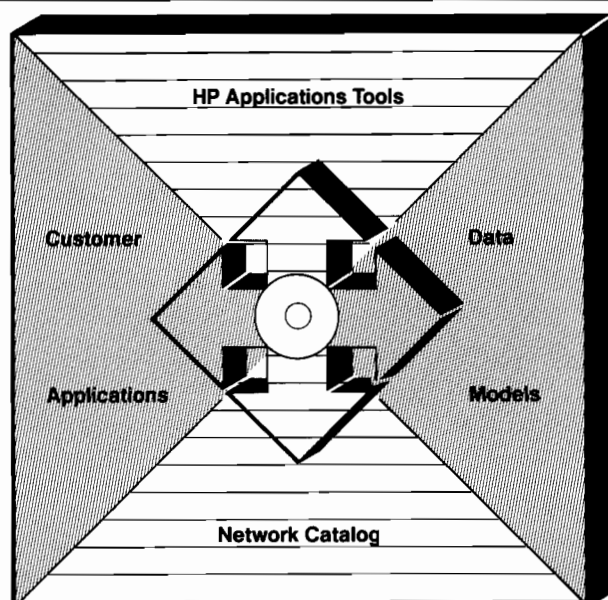
For MPE V Based and 900 Series HP 3000 Systems

HP System Dictionary provides a central information resource for documenting data, applications, programs, files, users, input forms, and network configurations for HP 3000 systems. It is a global dictionary which assists data administrators in effectively managing system and network resources. It also increases programmer productivity by providing timely and accurate system information for software development, while automating the generation of TurboIMAGE schemas and COBOL copylibs. As a component of HP's Distributed Application Technology it assists software developers and network administrators in documenting data, applications and hardware within a network. With the addition of compiled dictionaries and a copy/merge facility, System Dictionary can assist in managing distributed data and applications within a network.

System Dictionary plays a key role in HP's data management solutions. As a foundation technology in HP Information Management Architecture, it is an important component of HP's software development, reporting and data modeling solutions. System Dictionary can be accessed and updated by software productivity tools as well as by packaged or user-written application systems. System Dictionary is also an ideal resource for Business Report Writer/V, offering access to data within a system or a network for reporting purposes. Many companies want to establish a corporate data model and data standards. System Dictionary provides the framework for modeling virtually any data, application, organization or topology and assists in migrating to company-defined data standards.

Key Features

- **Entity-Relationship Model**
Provides a documentation structure closely matching the user's conceptual data model.
- **Extensibility**
Lets customers tailor the structure of the dictionary to fit their information needs.
- **Programmatic Access**
Allow programmers to directly access and update the dictionary through a set of documented intrinsics.
- **Compiled Dictionaries**
Fast, compact, read-only subsets of a System Dictionary which facilitate data sharing within a network.
- **Copy/Merge**
Allows versions of data definitions to be merged within one dictionary or to be merged into a System Dictionary residing on another node in the network.



System Dictionary is a Single Point of Reference for System & Network Information

- **Domains**
Handles name conflicts by putting conflicting sets of definitions in separate "name spaces".
- **Versions**
Allows various versions of a set of data definitions: "test", "production" and "archival" versions, to coexist separately within the same domain.
- **Synonyms and Aliases**
Documents alternate names used within the dictionary and by external systems.
- **Powerful User Interface**
Offers flexible formats, full abbreviations, on-line help, and a "redo" function. Log files, text files and user-defined macros can be used for entering data. Reports on the structure and contents of a dictionary assists in documenting data resources.
- **Localization**
Allows every user input, dictionary response, and all dictionary names to be converted to local languages.
- **Flexible Security**
Defines user security capabilities and allows customers to restrict access to particular domains and dictionary objects.
- **Dictionary/V Conversion Utility**
Automates conversion from Dictionary/V (formerly Dictionary/3000) to System Dictionary.

■ Database Utilities

Creates TurboIMAGE schemas and root files from dictionary definitions and produces dictionary definitions from a TurboIMAGE root file.

■ VPLUS Forms Definition Loader

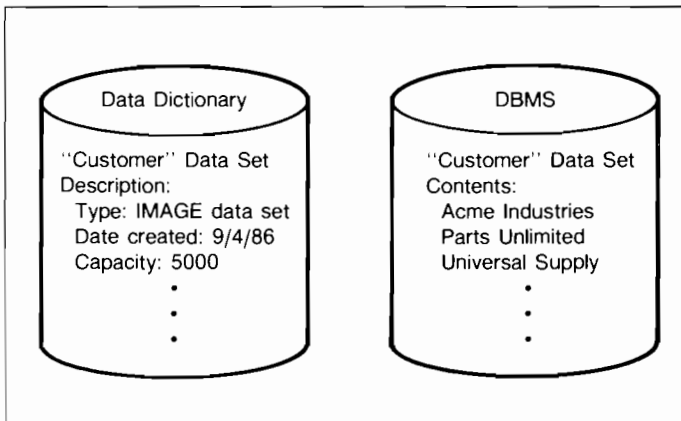
Loads information about VPLUS forms files into the dictionary.

What is a Data Dictionary?

A data dictionary stores information about a computer system, including descriptions of the databases, applications systems, account structures, as well as system and network configurations. This information is called *metadata*. The dictionary ensures that the metadata is entered in a standard format, is stored within a well-organized structure, and is easily available to users or programs that need the information.

Managing Metadata vs. Data

A data dictionary and a database management system are complementary products. A dictionary manages information about data (metadata) on a system or within a network, while a DBMS manages the data itself. For example, System Dictionary contains information about the format, usage and location of data. A database management system contains the data itself.



Comparison of the Information in a Data Dictionary and a DBMS

In addition to documenting TurboIMAGE databases, System Dictionary can describe MPE files, KSAM files, VPLUS forms files, programs, network device configurations, as well as other system and network components. It can model virtually any kind of data and can document the relationship between these components.

System Dictionary Applications

System Dictionary can be helpful to data administrators, application developers, and anyone else who is responsible for developing or maintaining application software.

Better Documentation

System Dictionary has many advantages over paper-based documentation methods:

- It is more accurate and consistent since dictionary objects are documented once, rather than once for each system using them.
- It is more timely and convenient since users always have on-line access to the most recent versions of the documentation.
- It is easier to work with since users have one source for all the system documentation they need.
- It is more reliable due to a tighter integration with the systems being described. The System Dictionary utilities and the programmatic access features provide the primary sources of this tighter integration.

Promoting Data Standards

Creating and enforcing standards is essential to effective system data resource management. If the applications on a system use different sets of names and formats, sharing information among them becomes extremely difficult. System Dictionary can be a powerful tool for promoting data standards in the following ways:

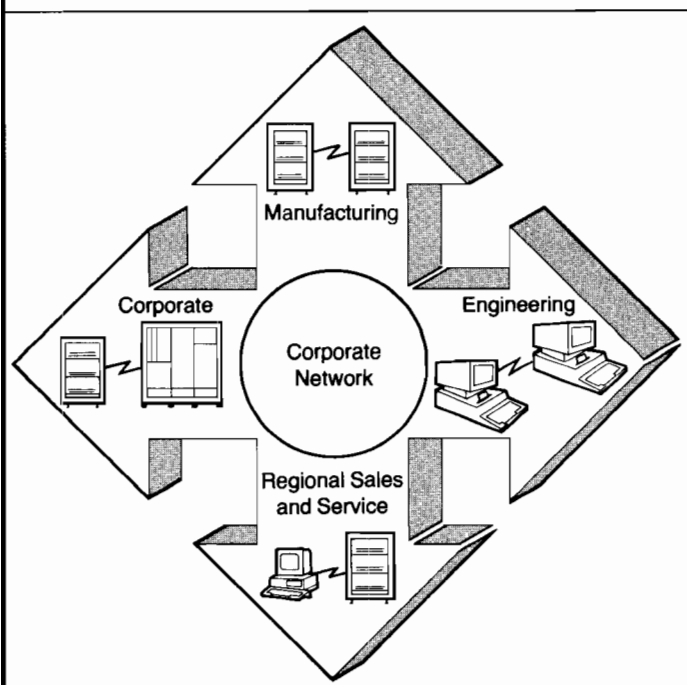
- Programmers can easily find the standard names and formats for data elements as they include them in new programs.
- Programmers can be prevented from using names that are already being used elsewhere to mean something else, thereby preventing naming conflicts.
- New definitions can be reviewed by the data administrator for consistency with system standards.
- Standard definitions can be shared with other nodes in a network via compiled dictionaries which can be sent to other nodes. On the other network nodes the compiled dictionaries can provide standard data definitions to the local applications whether or not a "Master" dictionary resides there.

A Network Catalog

The System Dictionary can be used to store the following information about a network:

- The location of all application systems, files, databases, programs, and devices in the network and the relationships between them.
- The network topology including names, locations and descriptions of network nodes and other hardware including the access paths which connect them.

This network information can be used by network administrators in managing a network, as well as by application programmers in writing distributed applications which will run in a networked environment.



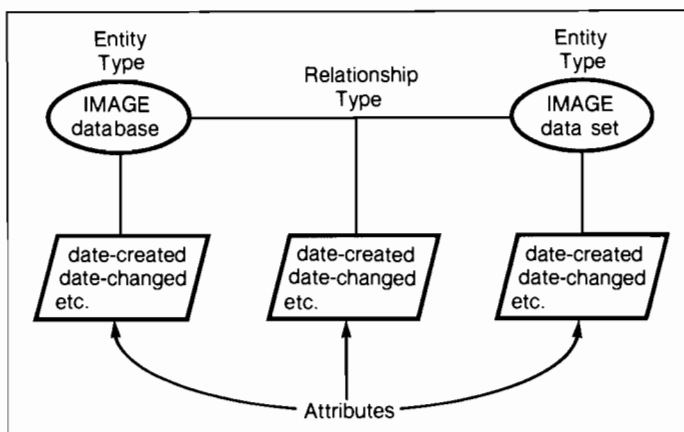
System Dictionary is a Valuable Catalog for Hardware and Software in a Network

Other Dictionary Uses

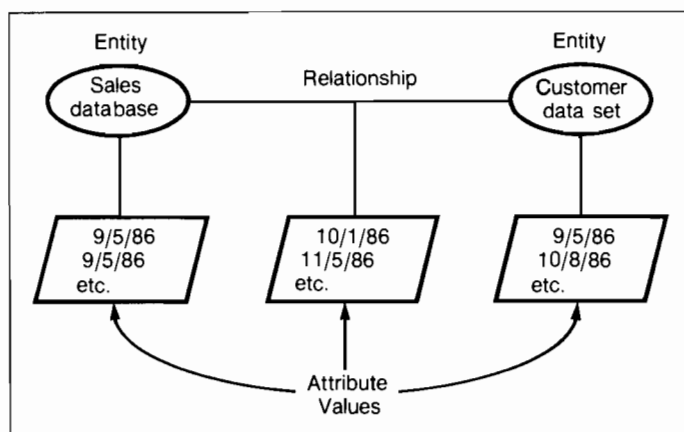
- Prior to making naming or format changes, users can consult the dictionary to see which applications systems will be affected.
- After naming changes, the old names can be documented as synonyms or aliases. This makes adjustments to new naming schemes easier by allowing users to continue to access data using the old names.
- System Dictionary can be used to model complex data and programming structures for existing systems and systems under development.
- System Dictionary can cut waste and redundancy by helping users locate programs and data that they otherwise would have developed independently.
- The Database Creation Utility automatically generates TurboIMAGE schemas and root files from dictionary definitions.

Entity-Relationship Model

System Dictionary is organized according to the entity-relationship model. Entities represent all objects described in the dictionary (e.g. files, records, forms, documents, devices, users etc.). Relationships describe the ties between these objects that give the system its structure. Relationships can tie together as many as six entities, although most only involve two. Attributes can be added to both entities and relationships to provide descriptive data.



System Dictionary Structure



System Dictionary Content

A core set of predefined entity and relationship types is provided which allows users to document IMAGE, TurboIMAGE, HP SQL, KSAM and MPE files, VPLUS screens, COBOL programs, and network device configurations. Additional attributes, entity types, or relationship types can be added by the user as needed through the extensibility feature.

Design Features

Extensibility

Customers can extend the core set to add new entity types and relationship types in order to customize the dictionary to reflect their own data model. New or existing attributes can also be added to existing entity types and relationship types. The new entity types, relationship types, and attributes are treated exactly the same as the predefined core set for inputs, outputs, and all other dictionary operations. With extensibility, customers can modify the dictionary to fit their data model instead of modifying their data model to fit the dictionary.



Programmatic Access

The System Dictionary intrinsics make it possible for customers and independent software vendors to modify application packages so that they automatically update or refer to the dictionary. Using these intrinsics, they can also design special purpose user interfaces and utility programs to meet their particular needs. Since these intrinsics are the same ones called by the user interface, customers have programmatic access to the full set of System Dictionary functions. The System Dictionary intrinsics provide yet another way for customers to tailor the dictionary to fit their needs.

Synonyms and Aliases

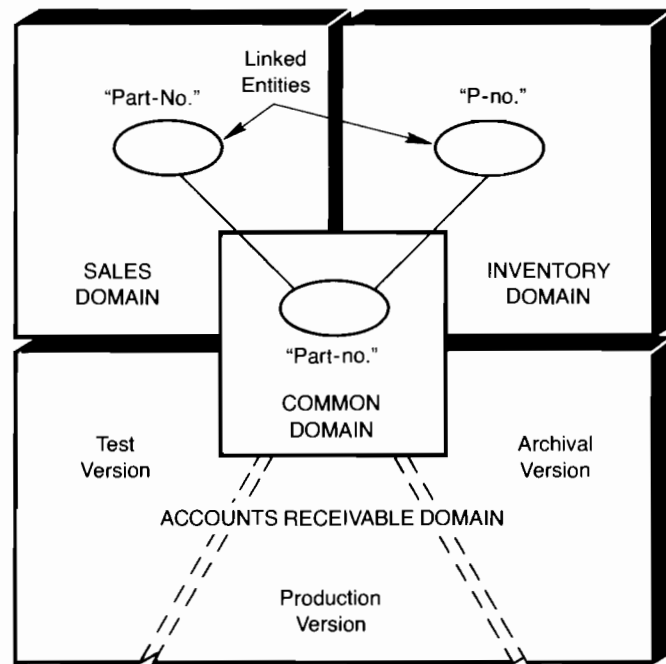
Synonyms are alternate entity names which can be used within the dictionary to refer to the entity for modifications or reports. Users can define any number of synonyms for an entity. Aliases are attributes that specify names used outside of the dictionary to refer to a dictionary entity. For example, in COBOL programs, a data set might have a different name from the one used in the dictionary. This would be recorded in the dictionary as the data set's COBOL-alias. With synonyms and aliases, documentation for application systems with incompatible naming schemes can be integrated within a single dictionary.

Easy Localization

System Dictionary's system of internal and external names and its use of the Native Language Intrinsics allow every input and response, as well as all dictionary names to be converted to local languages. Native Language Intrinsics allow users to localize every word that appears on the screen. External names allow the use of localized names for all dictionary objects. At the same time, internal names provide unique, permanent names that can be counted on by application developers to be identical for all System Dictionary installations.

Domains

Often, when an existing application system is documented within the dictionary, only a subset of the definitions need to be shared with the rest of the system. Domains provide a way to share global definitions throughout the dictionary while keeping application-specific definitions partitioned within a local domain. Definitions are shared by creating links from the local domain, where they were originally entered, to the common domain where global definitions are documented. With domains, naming conflicts between application systems can be handled by documenting the applications in separate domains.



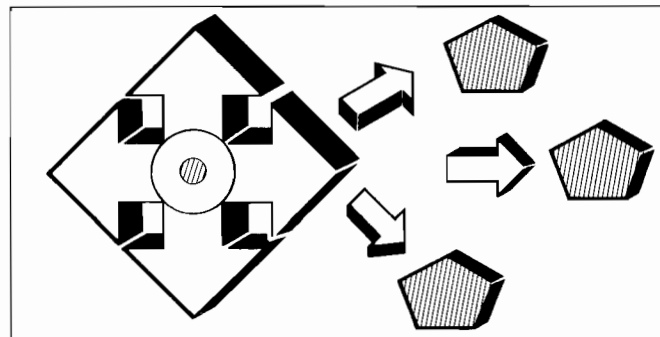
Domains and Versions

Versions

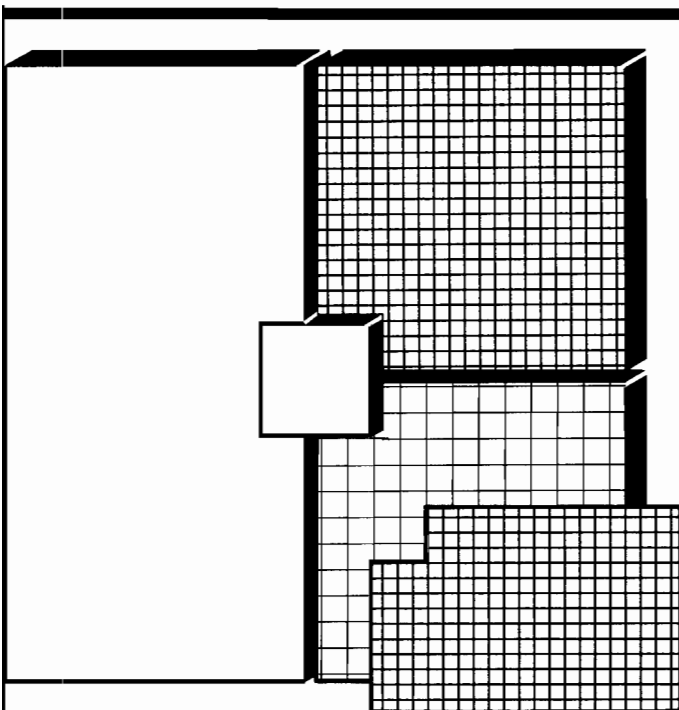
Each domain can have multiple versions. Several versions can be assigned "test" or "archival" status while only one at a time can have "production" status. New test versions can be created through the utilities or by copying existing versions and modifying them. To protect production and archival versions from unexpected changes, modifications to these versions are not allowed. Users desiring to implement version control with System Dictionary can make required changes by switching the version status to "test".

Compiled Dictionaries

Versions within a System Dictionary can be compiled into a compact, read-only file format called a "compiled dictionary". A compiled dictionary can be used by any System Dictionary application for which read-only access is sufficient. The compiled dictionary format improves the run-time performance of dictionary retrievals. [This functionality not available on the 900 Series at first release.]



Compiled Dictionaries are Read Only Subsets of a System Dictionary



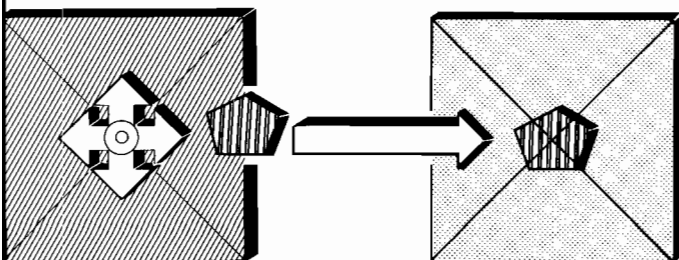
Copy/Merge definitions from one Version within a Dictionary to another Version

Copy/Merge

Versions within a System Dictionary can be merged in order to combine two sets of data definitions or to update an existing version without manually re-entering the definitions. Dictionary structure and security can be merged along with metadata to ensure consistency.

A version of data definitions, as well as structure and security, can also be shared with a System Dictionary residing on another node in the network. In order to do this the version to be shared is compiled. In compiled dictionary form it is sent to a remote system and merged into the System Dictionary which resides there. Before implementing the merger of two versions a customer can preview the effects of the changes on the resultant version in order to assure its success.

When proliferating a corporate data model throughout a company, compiled dictionaries and copy/merge capability allow subsets of a System Dictionary to be shared with multiple locations and updated as necessary. [This functionality not available on the 900 Series at first release.]



Compiled Dictionaries can be used to Merge Structures and Definitions which Exist on Different Systems

Flexible Security

Security scopes determine the level of access granted to specified dictionary users. Up to 128 separate scopes can be defined. A particular scope can be assigned any combination of capabilities to extend the dictionary, define other scopes, define domains, define versions, or create, modify, and read entities and relationships. Users with responsibility for entities, relationships, and domains can also control access to them by others. The flexibility of System Dictionary security permits customers to establish the appropriate level of security in instances where it is needed, while permitting easier access where strict security is not required.

User Interface Features

The user interface is command-driven with a flexible format. A "redo" command (similar to the MPE "redo" command) allows users to correct errors with a minimum of retyping. Abbreviations and on-line help are also provided. Commands can be entered from the keyboard, a text file, log file or user-created macro.

Users can produce reports on both the structure and content of the dictionary and save report definitions to run again later. For convenience, several pre-defined reports are also provided. Reports can search by entity names and attribute values using wild card characters and linking any number of values with operators like "and", "or", "<", and "=".

System Dictionary Utilities

The following utilities are included with the purchase of System Dictionary:

The **DICTIONARY/V CONVERSION UTILITY** helps Dictionary/V customers to convert existing dictionaries to domains within a System Dictionary without manually reentering data.

The **DATABASE CREATION UTILITY** creates an **IMAGE** schema and root file from definitions in the dictionary. The **DATABASE DEFINITION LOADER** loads database structure information into System Dictionary from an **IMAGE** root file. These utilities greatly reduce the amount of data entry time required for the dictionary. Both utilities work with **IMAGE** and **TurboIMAGE** databases and can be run either interactively or in batch mode.

The **VPLUS FORMS DEFINITION UTILITY** loads information about **VPLUS** forms into the System Dictionary.

In addition, customers may wish to purchase the **COBOL DEFINITION EXTRACTOR** which is sold as a separate product (#32255A on MPE V based systems and #32257A on MPE XL based systems) and can be used to generate COBOL data definitions for COBOL copylibs from definitions in the System Dictionary.

System Environment

System Dictionary and the System Dictionary COBOL Definition Extractor are available on MPE V based HP 3000 systems (T-Delta-4 MIT or later) and on the 900 Series of HP 3000 systems.

Ordering Information

The right to use System Dictionary or the System Dictionary COBOL Definition Extractor on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Software

The following software products are available on MPE V based systems:

Product Number	Product Description
32254A	HP System Dictionary/V Right-to-Use product for MPE V based systems
32254R	HP System Dictionary/V Right-to-Copy product for MPE V based systems with sublicense
32255A	HP System Dictionary/V COBOL Definition Extractor Right-to-Use product for MPE V based systems
32255R	HP System Dictionary/V COBOL Definition Extractor Right-to-Copy product for MPE V based systems with sublicense

The following software products are available on the 900 Series of HP 3000 systems. System Dictionary/XL is also available at a discounted price as part of the Preconfigured 900 Series systems:

Product Number	Product Description
32256A	HP System Dictionary/XL Right-to-Use product for the 900 Series of HP 3000 systems
32256R	HP System Dictionary/XL Right-to-Copy product for the 900 Series of HP 3000 systems with sublicense
32257A	HP System Dictionary/XL COBOL Definition Extractor Right-to-Use product for the 900 Series of HP 3000 systems
32257R	HP System Dictionary/XL COBOL Definition Extractor Right-to-Copy product for the 900 Series of HP 3000 systems with sublicense

Documentation

The following manuals are provided with System Dictionary:

Part Number	Document Description
32254-90001	HP System Dictionary/V SDMAIN Reference Manual
32254-90002	HP System Dictionary/V Intrinsics Reference Manual
32254-90003	HP System Dictionary/V Utilities Reference Manual
32254-90004	HP System Dictionary/V General Reference Manual 1
32254-90005	HP System Dictionary/V General Reference Manual 2
32256-90001	HP System Dictionary/XL SDMAIN Reference Manual
32256-90002	HP System Dictionary/XL Intrinsics Reference Manual
32256-90003	HP System Dictionary/XL Utilities Reference Manual
32256-90004	HP System Dictionary/XL General Reference Manual 1
32256-90005	HP System Dictionary/XL General Reference Manual 2

One manual is provided with the System Dictionary COBOL Definition Extractor:

Part Number	Document Description
32255-90001	HP System Dictionary/V COBOL Definition Extractor Reference Manual
32257-90001	HP System Dictionary/XL COBOL Definition Extractor Reference Manual

Customer Training

A copy of the self-paced customer training course and HP's data dictionary primer is included free of charge with each copy of System Dictionary. Additional copies of these publications can be ordered separately:

Part Number	Title
22855A	HP System Dictionary Self-Paced Customer Training
5958-8527	Data Dictionaries Managing Information Networks

Information Access

Improving Business Decisions through Better Access to Information

Information Access is a unique information management tool and a component of Office Productivity Services. It is an information server that extends the reach of your data processing systems by delivering key business data throughout your organization. Information Access can provide the means for your people to make better decisions based on up-to-the-minute information wherever it resides. Additionally, by centralizing the administration of information distribution using high-security controls, Information Access can improve the manageability of essential data processing systems.

Easy Access for the End User — Higher MIS Productivity

Information access increases the productivity of your end users by providing secured access to vital, current information for immediate and informed decisions. By choosing either interactive or batch access to information, users can determine the best use of their time and resources. The flexibility of management decision-making is increased because ad hoc analysis of up-to-the-minute information is now possible.

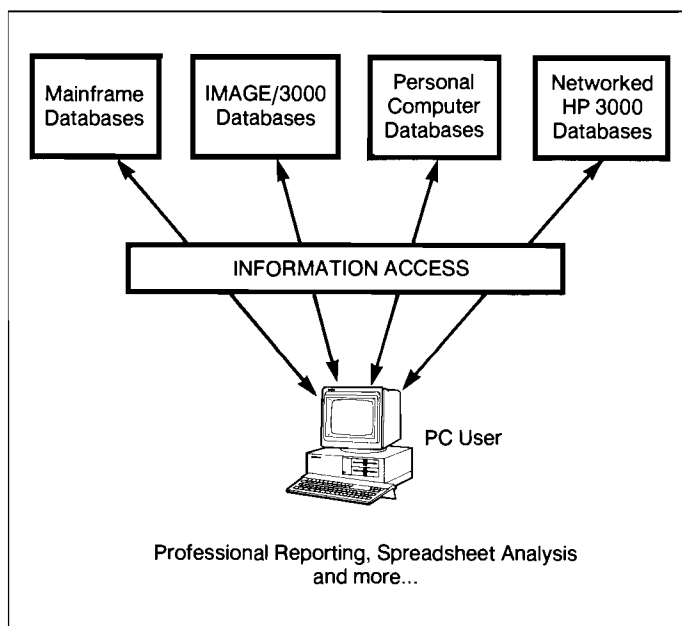
MIS professionals can also use Information Access to distribute information in a variety of file formats without generating custom programs. What has often required customized data processing reports can now be accomplished by end users with simple yet powerful menu-driven commands.

Access to Information wherever it Resides

Information Access provides the most current information to PC users in your organization by allowing access to PC data as well as host and remote IMAGE/3000 data. Mainframe data is also accessible through Cullinet's® Information Center Management System™ (referred to as C/ICMS). The user does not need to know where the information is located or any data structure commands.

Better Decisions

Once data is selected, Information Access helps organize your information for better decision making. The user now has the power to manipulate, cross tabulate, and reformat data. Professional reports can be generated or results can be



Improving Business Decisions through Better Access to Information

output directly to popular PC applications such as Lotus® 1-2-3® and R:Base 5000®. All data transfer and reformatting is automatic and transparent to the end user. The decision-making data your people need is where it belongs — in the hands of the user.

Central Administration for Security and Control

Information Access can be the key to your information management and security needs. The powerful administrator utility allows central organization of accessible information by creating logical views of data from host, networked, or mainframe databases. The database administrator determines the specific mainframe databases, IMAGE/3000 databases, data sets, and even specific data items that individual personal computer users can access. For mainframe access, the Information Access administrator utility provides a tailorable, scriptable logon mechanism to fit your specific security needs. Cullinet's C/ICMS security is maintained and enhanced. This administration capability, combined with the standard IMAGE/3000 and MPE security, guarantees the integrity of your database management system.

Features

- The **Database Administrator** utility provides control over the access and distribution of database information by centralizing its administration. The administrator utility is menu-driven and simple to learn. It is used to define relational tables of data and to configure which PC users can access the data, ensuring maximum control for your organization.
- **Security** can be placed on databases, data sets, and even data items. User groups and passwords are set only by the database administrator. Sensitive mainframe logon information, such as user identification and password security, is also protected. All logon information is encrypted and kept in privileged files on the HP 3000. In addition to the extensive control provided by standard IMAGE/3000 and MPE security, mainframe data is protected by Cullinet's C/ICMS security mechanisms. Because Information Access security is layered on top of each of these controls, your organization can be assured data integrity will be maintained.
- **Host processing is offloaded** because the unique software architecture of Information Access intelligently combines PC, HP 3000 and mainframe processing power to best use your organization's computing resources.
- **Programmer productivity** is improved by allowing end user access to information that previously required ad hoc reports from your programming staff. Application development can be accelerated since Information Access, used as a prototyping tool, can quickly combine information from a variety of sources.
- **Menu-driven** and designed as an end user application, Information Access is easy to learn and requires minimal training. Help screens and self-paced training are included.
- A **tabular view of data** is provided so end users do not require knowledge of data structure or data location. Users can view a combination of data from multiple mainframe, IMAGE/3000 and personal computer databases by simply selecting a menu item.
- **Status messages** estimate the length of time to process data inquiries to host and remote IMAGE/3000 databases. Your end user is notified of the length of time the PC will be dedicated to the data inquiry and what percentage of the inquiry is complete. Access requests for mainframe data receive status messages indicating elapsed process time and the number of records selected. A cancel key can be used to terminate the request effortlessly if circumstances dictate.
- **Batch capabilities** allow data access requests to be run as a batch job. Host batch, initiated from the HP 3000 by the database administrator, allows off-hours processing of data facilitating regular consolidation of information for workgroup needs. PC batch allows the Information Access user to use the PC for other purposes while an extensive query is processing. At a later time, the user can return to Information Access and inquire about the status of the job. If complete, the user can then access the resultant data.

- **Automatic conversion** of key business data into many popular formats is accomplished in one easy step. Information Access reformats PC, IMAGE/3000, and C/ICMS information for use in popular applications such as Lotus 1-2-3, R:Base 5000, d:BASE II®, C/ICMS tables, or popular word processing and graphics packages.
- An **upload capability** allows PC data to be transferred to the HP 3000 and saved in a variety of standard formats. PC or IMAGE/3000 data can also be transferred to the mainframe and stored in a Cullinet C/ICMS format.
- The **built-in personal report writer** allows generation of both ad hoc and routine reports. This report writer is as easy to use as Lotus 1-2-3, yet as powerful as you need — including complex statistical and cross tabulation functions. The personal report writer is menu-driven and allows preprinted forms and custom templates for repeated use.

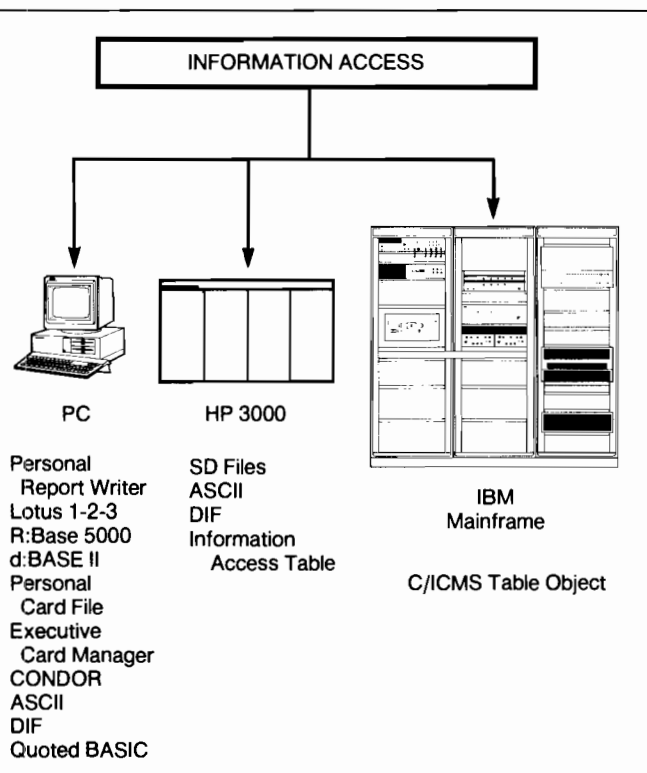
Wide Range of Accessible Data

- Mainframe data residing in Cullinet's Information Center Management System as well as other data sources supported by Cullinet's Information Center Management System e.g., VSAM and, through additional Cullinet facilities, IMS and DL/1
- IMAGE/3000 databases on the host HP 3000
- IMAGE/3000 databases on networked HP 3000s
- R:Base 5000
- d:BASE II
- Personal Card File
- Executive Card Manager
- CONDOR™

Multiple Output Options from the PC

Personal Report Writer

- PC Applications:
 - Lotus 1-2-3
 - R:Base 5000
 - d:BASE II
 - Personal Card File
 - Executive Card Manager
 - CONDOR
- PC File Formats:
 - DIF
 - ASCII
 - Quoted Basic
- HP 3000 MPE Formats:
 - ASCII
 - SD
 - DIF
 - Information Access Table
- Mainframe Formats:
 - C/ICMS Table Object



Information Access Output Options for PC and IMAGE/3000 Data

Product Information

Hardware Environment

Personal Computer Systems

- HP Vectra
- HP Touchscreen
- IBM® PC™/XT™/AT™

All models require:

- Dual disc drive (double sided at a minimum)
- 512K bytes minimum memory
- 640K bytes minimum memory with LAN

Mainframe Systems

- Systems that support Cullinet Information Center Management System and HP IMF/3000 or SNA IMF e.g., IBM 370, IBM 30XX, IBM 43XX, or compatibles

IBM Operating Systems:

- MVS
- MVS/SA

HP 3000 Systems

- MICRO 3000
- MICRO 3000XE
- Series 37
- Series 39
- Series 40/42
- Series 44/48
- Series 52
- Series 58
- Series 64/68
- Series 70

All models require:

- 2M bytes minimum main memory
- 4M bytes minimum main memory with LAN
- 1M bytes additional main memory recommended with IMF/3000 or SNA IMF

Networking Environment

PC to HP 3000 connection requires either:

- ATP RS-232-C or RS-422-C connection for one of the following:
 - Dedicated
 - Dial Up
 - X.25 via HP 2334A Plus — X.25 Multiplexer
- INP
 - X.25 via HP 2334A Plus — X.25 Multiplexer

or:

- HP Officeshare Networking Family
 - HP ThinLAN for PCs
 - HP StarLAN for PCs
 - HP SERIAL Network

HP 3000 to Mainframe connection requires one of the following:

- IMF/3000
- SNA IMF

Software Environment

Personal Computer Systems

- HP Vectra Personal Computer Systems
 - MS DOS (Version 3.10 or later)
- HP Touchscreen System
 - MS DOS (Version 2.11 or later)
- IBM PC/XT Systems
 - PC DOS (Version 2.11 or later)
- IBM PC/AT Systems
 - PC DOS (Version 3.10 or later)

Mainframe Systems

- Cullinet's Information Center Management System Release 1.1 Final
 - TAPE VSN C8607M or later

HP 3000 Systems

- MPE V/E (G.02.03 or later)

Ordering Information

Product Number	Product Description
32596A	Right-to-Use Information Access Kit For host and one to ten PCs
32596R	Right-to-Copy Information Access For host and one to ten PCs
32596L	Information Access License Upgrade For one to ten additional PCs
32596T	Information Access PC Documentation Five extra copies
32598A	Information Access Cullinet Link
32598R	Right-to-Copy Information Access Cullinet Link
36896A	Right-to-Use Information Access Remote IMAGE/3000 Link
36896R	Right-to-Copy Information Access Remote IMAGE/3000 Link

Support

Hewlett-Packard offers a broad range of support services and, with over 180 service and support facilities throughout the world, can meet the special needs that exist in your office. HP OFFICE-ASSIST: Information Access is a recommended implementation service that can help you introduce the Information Access product successfully into your organization. The product number for HP OFFICE-ASSIST:Information Access is 32596A + 60A.

Company or product names followed by ® are U.S. registered trademarks of the respective company. Company or product names followed by ™ are registered trademarks of the respective company.

Cullinet and Information Center Management System are trademarks of Cullinet Software, Inc.

IBM, PC, XT, and AT are trademarks of International Business Machines Corp.

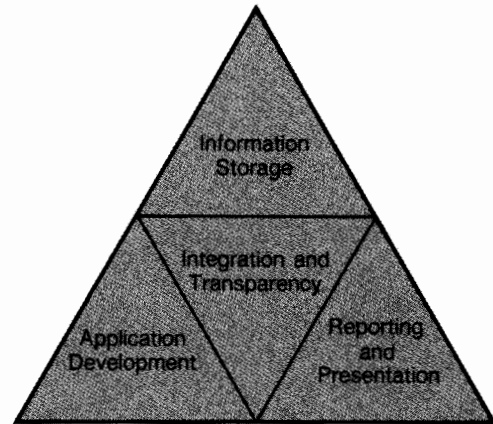
Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation.

d:BASE II is a U.S. trademark of Ashton-Tate.

R:Base 5000 is a U.S. trademark of Microrim, Inc.

CONDOR is a trademark of Condor Computer Corp.

System Management Tools



HEWLETT-PACKARD

Product Number 30167A

TurboSTORE/V

A High Speed Backup Program for
MPE V Based HP 3000 Systems

Key Benefits

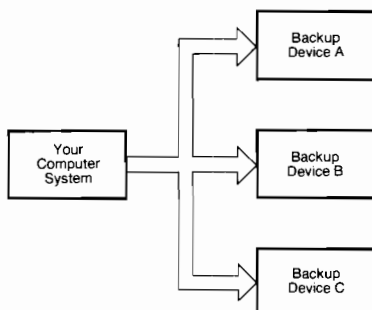
- Cuts backup time in half
- Increases system availability for batch/online activity
- Reduces the time required to recover data
- Easy to learn and operate: user-interface and documentation tightly integrated with MPE-V/E STORE/RESTORE

Key Features

TurboSTORE is a full functionality backup program that eliminates bottlenecks associated with system backup. As a result, it provides rich performance benefits for a wide array of HP 3000 systems, disc drives and tape drives.

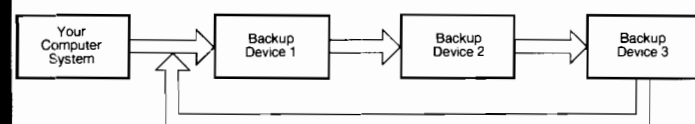
Multiple Tape Drive Use in Parallel

TurboSTORE supports simultaneous backup to multiple tape drives. This eliminates any bottlenecks created by backup device speed limitations. Up to 4 tape drives can be used in parallel cutting backup time substantially.



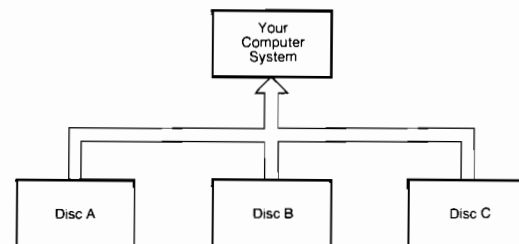
Multiple Tape Drive Use Sequentially

TurboSTORE also provides the ability to switch automatically to a second tape drive when the tape on the first drive has been completely written. This eliminates inactive time caused by reel rewind and load/unload. Up to four backup devices may be used in sequence in a round-robin fashion.



Merged Transfers From Multiple Discs

File and disc access is the most common backup bottleneck faced by configurations that include a more powerful CPU and a high speed tape drive. TurboSTORE increases the effective disc access rate over conventional backup software by merging transfers from multiple discs into a single, high volume, data stream.

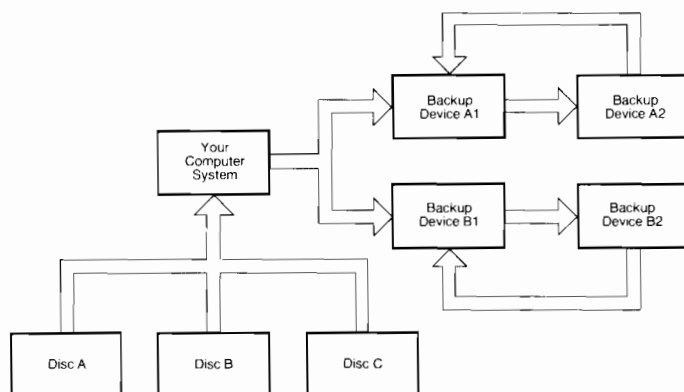


This higher data volume can be used to better utilize the potential of a single high speed tape drive or, more typically, drive multiple tape drives.

Combining Features for Maximum Performance

The full power of TurboSTORE will usually be realized by using combinations of the features described above. The increased file system output obtained by merging transfers from multiple discs can be used to drive multiple tape drives in both the above mentioned modes.

For example, customers can use merged transfers from multiple discs to drive two tape drives in parallel. This is the most effective option when only two tape drives are available. Another example, as shown below, TurboSTORE can operate two tape drives in parallel and automatically switch to a second pair of drives when tapes on the first set have been written.



Integrated with Store/Restore for Maximum Ease of Use

The TurboSTORE user-interface has been designed as an evolutionary superset of the STORE/RESTORE capability provided by MPE-V/E. The TurboSTORE command syntax is exactly the same as the STORE/RESTORE utility except for three new parameters: STORESET, RESTORESET and INTERLEAVE. If these parameters are not specified, TurboSTORE will function exactly like MPE-V/E STORE/RESTORE. In fact, TurboSTORE is installed as a replacement for STORE/RESTORE in order to ensure that the customer does not even need to use the RUN command to invoke TurboSTORE.

As a result, very little additional training is required to use TurboSTORE and functionality can be turned on or off with great ease at the users' convenience.

Moreover, TurboSTORE documentation has been rolled into the MPE-V/E manuals "Storing and Restoring Files (32033-90133)" and "MPE-V/E Backup and Disaster Recovery (32033-90134)". This ensures that all the information required to operate backup and recovery exists in one location for ease of reference.

Performance

Most customers can expect a 100% improvement in performance when two tape drives are used. This results in time savings of 50% of the total time required to backup.

The precise reduction in backup time is dependent on several factors including:

- number of tape drives used
- available CPU
- tape drive speed
- number and size of disc files
- distribution of files across discs

Recovery performance is improved by 10 - 20% when TurboSTORE tapes and a single device is used. Additional time savings can be obtained by using multiple devices in sequence.

Money Back Guarantee

Hewlett Packard guarantees your satisfaction with this product. For the few customers who, because of very special circumstances, do not obtain adequate performance, TurboSTORE comes with a money back guarantee.

Customers have the option of returning TurboSTORE to Hewlett-Packard within 60 days of installation and get a full refund if the product does not meet their expectations.

Compatibility

When the 'merged transfer' or 'parallel device' capabilities are used a new tape format is generated. This format can only be read by those systems that have TurboSTORE installed. If these two features are not used, TurboSTORE generates tapes in the regular MPE-V/E STORE/RESTORE format and can hence be read on systems without TurboSTORE.

System Environment

- TurboSTORE is supported on V-MIT and subsequent releases of MPE-V/E.

Ordering Information

The right to use TurboSTORE/V on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Product Number	Product Description
30167A	TurboSTORE/V High Speed Backup Program
30167R	Right-to-Copy TurboSTORE/V

Documentation

Part Number	Document Name
32033-90133	Storing and Restoring Files
32033-90134	MPE-V/E Backup and Disaster Recovery

HP Security Monitor/V

A System Security Program for MPE V Based HP 3000 Systems

Key Benefits

- Protects system resources and sensitive data
- Monitors security breaches
- Prevents unauthorized access
- Can be tailored to fit specific system environments
- Eases transition between old and new security practices

Key Features

HP Security Monitor is a fully integrated system security program that protects both system resources and sensitive data from unauthorized access. Building on the strong security of the base HP 3000 systems, it allows improved Password Protection, Stronger Audit Trails, and Tighter Terminal Security.

Improved Password Protection

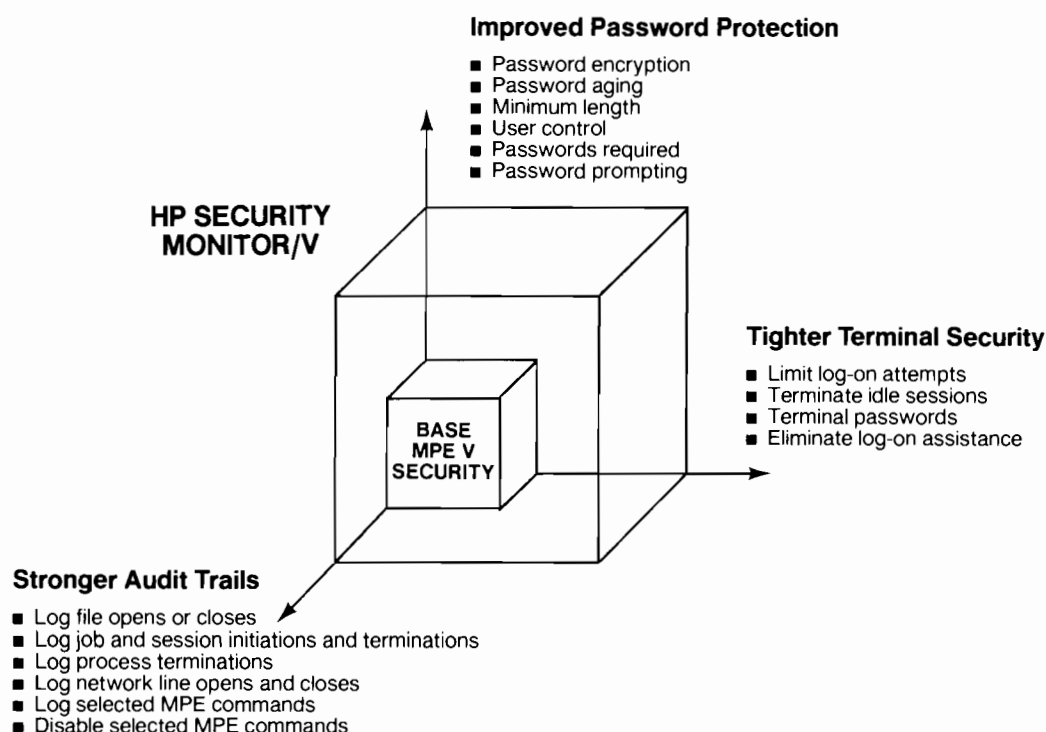
Passwords provide the first line of defense against unauthorized access to system resources and data. While these passwords are optional on base HP 3000 systems, with HP Security Monitor, the Security Administrator can require all users to supply a password before gaining access to the system. The Security Administrator can also impose password minimum length and maximum age requirements.

Longer passwords are more difficult to guess while password expiration forces the user to essentially renew security by changing passwords.

Through the PASSWORD command, HP Security Monitor also allows users to take control of their own passwords. This eliminates the need for a centrally administered list of passwords, which can be compromised.

The Security Administrator can also reject embedded passwords on interactive log ons, forcing the user to wait for the password prompt, and insuring that the password won't be echoed back to the user. This feature has the added benefit of sensitizing the user as to the importance of keeping passwords secret.

Lastly, HP Security Monitor can encrypt passwords, based on the highly secure one way Data Encryption Standard. This feature ensures that passwords are only stored on the system in non-readable form. "One way" refers to the fact that even when armed with both the algorithm and the encrypted text, an intruder couldn't deduce the original password.



Stronger Audit Trails

Audit trails allow security conscious customers to monitor and correct security breaches in the system. HP Security Monitor has improved audit trails by giving the Security Administrator the ability to log a wide variety of specified security relevant events. These events include: file opens, file closes, job and session initiations and terminations, process terminations, and network line opens and closes.

HP Security Monitor also allows the Security Administrator to selectively disable specified MPE commands. This feature can be used to isolate general users, allowing them access to the system while preventing their access to more powerful functions. HP Security Monitor can also monitor the execution of these MPE commands prior to disabling them, thereby identifying programs which would need to be modified.

Tighter Terminal Security

Terminal security prevents unauthorized intruders from bypassing the log-on security. HP Security Monitor can automatically terminate all command interpreter sessions which have been left idle for a specified time period. This prevents unauthorized intruders from simply using a logged on terminal which has been left idle. Also, HP Security Monitor can automatically terminate sessions in which the log-on UDS fails. This feature is useful when the log-on UDC is being used to restrict users to a specific application program, and to prevent general access to the system. When such an UDC fails, HP Security Monitor immediately terminates the session, preventing the user from bypassing the log-on security precautions.

HP Security Monitor can add another layer of security by requiring passwords on individual terminals and by eliminating any log-on assistance after an invalid log-on attempt. For example, an intruder with a valid user password, a valid account password, and an invalid group password will not learn that the group password was invalid, he will only learn that the total log-on combination was invalid. Finally, HP Security Monitor can make a terminal unavailable to the intruder after a configurable number of invalid log-on attempts.

Smooth Implementation

Building on the strong security of base HP 3000 systems, the security features of HP Security Monitor can be gracefully integrated on an as needed basis. All of its features are optional, and each feature can be enabled individually.

HP Security Monitor is also designed to phase in features, thereby easing the transition from the old to the new security practices on the system. This design gives customers wide latitude to tailor system security to fit their particular needs, while at the same time providing a smooth implementation.

Compatibility

By design, some of the higher security features of HP Security Monitor prevent execution of some of the ease of use features in HP, customer, and third party software. Depending on the specific security features enabled, there are some situations where customer might want to modify their programs, jobstreams, and UDCs to take full advantage of HP Security Monitor.

System Environment

HP Security Monitor is supported on V-MIT and subsequent releases of MPE V/E.

Ordering Information

The right to use HP Security Monitor on an HP 3000 computer is available according to the terms and conditions of the Hewlett-Packard Software Purchase Agreement.

Product Number	Product Description
30392A	HP Security Monitor Provides operating system security.
30392R	HP Security Monitor Right to copy

Documentation

Part Number	Document Name
32033-90136	MPE V Account Structure and Security

Technical information in this document is subject to change without notice.