

RAPID/3000

Nancy Calwell
Hewlett-Packard

OVERVIEW

RAPID/3000 is a new product that has been introduced by Hewlett-Packard. The RAPID/3000 product package contains four components. Together, these components make up a powerful productivity tool when used in conjunction with existing HP3000 application development software. This product increases the productivity of both programmers and end users by effectively separating their needs for data. Thus, the database administrator and the end user are provided with a set of tools to solve each of their respective data processing needs.

The components of RAPID/3000 are broken down as follows:

- HP TRANSACT/3000
- HP REPORT/3000
- HP INFORM/3000
- HP DICTIONARY/3000
 - Dictionary/Directory
 - utilities

The remainder of this discussion will deal with a more comprehensive view of the four RAPID/3000 components.

TRANSACT/3000

TRANSACT/3000 is a high level programming language that allows access to all MPE data files. It has the ability to call subroutines written in COBOL, PASCAL, SPL, and even TRANSACT. The flexibility exists to call MPE intrinsics. TRANSACT supports all of the data types that exist on the HP3000. It also has a "built-in" interface to VPLUS. A single TRANSACT command has the ability to replace many VPLUS intrinsics that currently the programmer must code.

Because of the ability to quickly code solutions with TRANSACT, the area of "prototyping" is a natural for this language. Feedback can be given to the application designer before a lot of time is spent on a possible miscommunication.

There is a feature in TRANSACT/3000 that allows dynamic debug capabilities while designing or maintaining a program. System and program errors are returned to the program for local handling with no need to return to the system environment.

REPORT/3000

REPORT/3000 is a nonprocedural report writer. This means that the statements within the report may be coded in any order. REPORT has the capability to re-

trieve information from IMAGE, KSAM, and MPE files. Now simple reports can be produced by using the defaults of REPORT/3000. By using the more advanced commands, customized reports can be written.

INFORM/3000

INFORM/3000 is a menu driven report generator. It is geared towards the entire company by providing an ad hoc inquiry ability to even the most unsophisticated user. Additionally, INFORM/3000 allows a relational type access to any IMAGE, KSAM, or MPE file that is defined in the Dictionary.

DICTIONARY/3000

The Dictionary is an IMAGE data structure that contains information about your production data. It provides a single place to go for data physical attributes, responsibility, descriptions, and various other forms of documentation.

Information about your data can either be interactively entered or automatically transferred from existing IMAGE root files into the Dictionary. Reports on the Dictionary contents into the Dictionary contents can easily be generated by using Dictionary commands.

DICTIONARY/3000 supports user views of data. In addition to documenting the physical nature of the information system, the DICTIONARY also documents and makes available through Inform, relational views of the data.

Included in the Dictionary package is a set of utilities that assist in the maintenance of IMAGE databases.

A Data Dictionary and Directory Facility

Have any of the following situations happened to you?

You have been using a centrally maintained database. All of a sudden, a program that has been running successfully for six months fails. The reason? Over the weekend, the database administrator changed the definition of a data element. You were not notified.

You are responsible for a centrally maintained database. You are requested by the Information Systems Manager to change the definition of a data element. You proceed to run yourself ragged locating all of the users that will be affected by this change. After a week of playing Sherlock Holmes, the change is done.

Monday morning you find ten angry users who were not notified.

You spend months developing a series of programs only to find out that a large part of the data produced by the new system is already available.

You find that the space on the system is starting to be worth more than a troy ounce of gold. Upon investigation of the problem, you find there are several files that look conspicuously alike. The problem? Data redundancy.

The problems mentioned above are only a few of those that annoy and waste the valuable time of data processing people every day. **DICTIONARY/3000** is the answer not only to the problems above, but to many situations that become problems in the wonderful world of data processing.

What is **DICTIONARY/3000**?

A dictionary must supply the answers to the questions

Who . . . ?

What . . . ?

Where . . . ?

When . . . ?

WHO is responsible for the data elements and files.

WHAT is their physical properties.

WHERE is their location.

WHEN is their usage.

The Dictionary should not only be used as a reference document, it is a tool to simplify documentation. It contains the description and directory information of the data. The data itself is still managed by **IMAGE**, **KSAM**, and **MPE**. Additionally, **DICTIONARY/3000** may be used to describe systems, programs, subprograms, and procedures. It is also possible to document the entire corporate organization (i.e., an organizational chart). Data may be gathered into logical groups to quickly generate reports. This is done with the help of **INFORM/3000**. Also included with the purchase of **DICTIONARY/3000** is a set of high powered utilities that assist in the maintenance of not only the Dictionary but production **IMAGE** databases as well.

When Should **DICTIONARY/3000** Be Considered?

If you have a database, you should be considering **DICTIONARY/3000**. If you have a lot of data or a need

to minimize the cost of application modification or development you should be considering **DICTIONARY/3000**. If you have a need to quickly link the data that resides in more than one **IMAGE**, **KSAM**, or **MPE** file. If you truly want to stop data redundancy. If you truly want to stop programmer time redundancy. If you have a **HP3000**.

A Closer Look

This section will deal more with the specifics on **DICTIONARY/3000**. The following outline indicates the topics that will be covered with respect to **DICTIONARY/3000**, **IMAGE/3000** and other **MPE** files. The presentation will be slide/lecture and will span the full hour. Time will be given at the end of the presentation for a question and answer period.

I. **DICTIONARY/3000**

A. A set of powerful tools

1. Descriptive tool

- a. Composition of items
- b. What types of files use these items.

2. Logistical tool

- a. Who uses which files
- b. Where are these files located

3. Operational tool

- a. Databases can be created thru the Dictionary
- b. Dictionary entries can be created thru a database.

B. Dictionary Structure

1. Physical structure

2. Logical structure

- a. physical entities (i.e., items, files, security)
- b. entity linkages

C. Physical entities

1. What are all of the physical entities available in the Dictionary

2. How are the entities used?

- a. entities whose usage is program defined
- b. entities whose usage is user definable

II. **DICTIONARY UTILITIES**

A. What are they

1. A list of the utilities
2. A closer look at what they do

B. **IMAGE** utilities vs. **DICTIONARY** utilities

1. **IMAGE** utilities still needed
2. **IMAGE** utilities possibly replaced