

HOW PRODUCTION CAN EXIST WITH DEVELOPMENT WITHOUT GETTING 'SMASHED'

by John D. Baker, P. Engineer
Hewlett-Packard

PURPOSE

This paper is intended for those in general management with relatively little experience with the HP3000. This is a description of an accounting structure and a naming convention that we developed at Kingston General Hospital. It is also a description of just one of several approaches that may be taken by management, some being more appropriate than others in certain situations.

Like everything else in the computer business, the development of an accounting structure is evolutionary and will change to meet the needs.

BACKGROUND

My background consists of being a Professional Engineer who got involved with running a very small computer shop for a hospital. I then moved on to health care consulting with Ernst & Whinney. I was very fortunate working at E & W in that I could work in health care as well as data processing. Part of the data processing requirement, however, was to assist the accountants with their audits where computers were part of the financial reporting.

Through this experience, I have encountered many methods whereby development takes place concurrently with production. In most instances, especially in the smaller shops, the process of implementing software relied on a "scouts honour" approach. Very little assurance could be given to me that the user was aware that changes had or would be taken place. In many instances the user knew only after the users couldn't balance the books three weeks after a change was made to a program.

It is important from everyone's point of view (user, programmer, shareholder) that when programs are modified by data processing personnel, that all concerned are aware and that precautions are taken to minimize potential disasters that could take place.

Upon my arrival at Kingston General Hospital, I was greeted with the following:

- * A series III with 1 MB memory and 240 MB disk.
- * A financial system that crashed every month end and periodically during the month.
- * No data processing plan.
- * A potential law suit with the turnkey organization which supplied the hardware and software.
- * Queen's University (who processed some of our systems, the major being payroll) was converting their hardware from Burrough's to IBM. I didn't know about this until late February. The conversion was set for May 1st.

I was also given one major mandate, besides putting everything in order. This mandate was to spread computerization throughout the Hospital. In short, turn around the bad impressions left by a bad implementation and make them positive.

This has been done, however, it has taken almost two years. The process we used was to:

- * Establish a formal steering committee.
- * Organize operations.
- * Contain all new development.
- * Establish standards such as:
 - o Database Standards
 - o Application Development Standards
 - o Documentation Standards
 - o Accounting Structure and Naming Conventions

I started on my trek and didn't get very far. My operations personnel consisted of basic-

ly two people who were not technically oriented. I had one programmer who had taught himself COBOL. He too didn't know much about HP systems. The Programmer's Course and the System Manager's Course weren't really all that helpful to me either. I knew what I wanted but I didn't have the knowledge nor the funds to purchase the knowledge. The next step was to read the System Manager's Manual over and over. I think I spent more time sleeping than I did reading.

I made a discovery last year. I wasn't alone. There were a few people at the IHPUG held in Montreal that had similar dilemmas to my own. It was there that I picked up more information than at any other single source except for the HP manuals. After Montreal portions of the manuals actually made sense.

INTRODUCTION

The primary purpose of a standard accounting structure and naming convention within a computer system is to provide:

- * Access to as many users as possible.
- * Data integrity.
- * Data security.
- * Meaningful file names for systems personnel.

A properly designed accounting structure protects a user's data from being modified by anyone except approved users, whether planned or accidental. The naming convention in conjunction with the accounting structure will allow the programmer to develop and/or modify program and data files using copies of live data without disturbing live data in any way.

ACCOUNTS

There are several account types assigned on the system. These are:

- * SYS
 - o System Management
 - o Operating system and utilities
- * HPSUP

- o Hewlett Packard diagnostics

- * PR0000

- o HFMS production programs and run files

- * QUEENS

- o Programs and run files which were resident on the Queen's Computer System and which now reside on the HP

- * KGH

- o HFMS database, sequential files and internally developed files specifically related to HFMS

- * DATABASE

- o Database Manager's account providing maintenance and control of:

- . Database schemas
- . Data dictionaries
- . System, application and program naming schemas

- * LIBRARY

- o Specifically designed to hold development tools to be used by programmers that can be accessed while in other accounts.

* PRODUCTION ACCOUNTS

- o Production programs and datafiles not specifically mentioned previously.

This paper is primarily devoted to detailing the specific accounting structure design of production accounts. The other accounts are designed for specific uses and in many instances, eg. HPSUP, the structure is dictated by other sources. In other cases, eg. DATABASE and LIBRARY, the actual structure will be evolutionary and will be entirely dependent upon the tools developed or purchased over the next few years.

FILENAMES

To understand the accounting structure, you must understand the approach MPE (HP's

operating system) takes in naming files. A file name is defined as follows:

Filename.GROUP.ACCOUNT

This is a hierarchial structure. An account can have several groups and each group can have a multitude of files. Through this simple structure we can design a sophisticated accounting structure and naming convention that will allow us to meet our objectives.

The term accounting structure basically relates to the GROUP.ACCOUNT portion of a filename. The naming convention for programs and files relate to the Filename portion. The prime objective is to be able to have all files in one group within one account in order for the programmer to function without impacting live data.

PRODUCTION ACCOUNTS

All production accounts will be assigned the following groups:

GROUP -----	ACCESS -----	CONTENTS -----
PUB	Default	User's exclusive use
IMG	R,W,A,X,S,L:ANY	All databases for a specific account. Schemas and dictionaries to reside in the DATABASE account
SEQ	R,W,A,X,S,L:ANY	All sequential files
KSAM	R,W,A,X,S,L:ANY	All direct (KSAM) files
JOB	X:ANY;R,W,A,S,L:GU,AL	All job control (STREAM) files
DEV	R,W,A,L,S:GU,AL	All development programs, data, etc. Once tested and approved by the user, source, run files data, etc. can be transferred to the appropriate groups
USL	R,X:ANY;W,A,S,L:GU,AL	All production programs in object (non-executable) code
RUN	R,X:ANY;W,A,S,L:GU,AL	All production programs in executable code
SOURCE	R,W,A,X,S,L:GU,AL	All programs in source code for a specific account. Schema source and data dictionary source to reside in DATABASE
UDC	R,X,L:ANY;W,A,S:GU,AL	All account UDC's

DOC R,X:ANY;W,A,S,L All documentation related to
:GU,AL production runs

FILE SECURITY

There is an additional aspect to file ownership. This involves the use of user names. User names are assigned to accounts that are generic and relate basically to the function of a particular user. Passwords are assigned to the user. Passwords are NOT assigned to the account. By having passwords assigned to the account, all passwords would have to be changed on all JOB files and databases should anyone leave the Hospital. In the case of HFMS, this would mean approximately 40 man-hours to make the necessary changes. This is an impossible task given our present

resources and software tools. By assigning a specific user name in a generic form, we can either change the password with relative ease or eliminate the user name from the accounting structure without fuss or bother. Our key concern is personnel having System Manager capabilities. This, however, will always be a concern no matter what security system is put in place.

USER NAMES

The following user names are assigned to each account and database designed by KGH:

USER NAME -----	CAPABILITY -----	HOME GROUP -----
MANAGER	Account Manager	PUB
MGR	Account Librarian	PUB
DATA0	Default	PUB
DATA1	Default	PUB
.	.	.
DATA9	Default	PUB
SUPERO	Default	PUB
.	.	.
PROG0	Account Librarian	DEV
.	.	.
PROG5	Account Librarian	DEV

Capabilities as defined in the above chart relate to capabilities within the account, not the programs that are being designed by Systems personnel. Specific capabilities within the actual application software are assigned within the databases and data dictionaries.

Each user is assigned to a Home Group meaning that every time a user sign's on to the system, he or she will have access to the files within that group. All users except PROG's are assigned to the PUB group. Applications designed by us are protected in other groups but are usable by all users of PUB.

All groups also have passwords assigned to them. All users have the capability to sign on to any group, however, should a user try to sign on to group other than their Home Group, the user must provide a password. Should the user sign on in the normal fashion, the user

will be prompted for only the user password. When signing on to the system, a user does not have to provide a group password, unless the user is signing on to a group other than the Home Group assigned. In this fashion, we install hurdles for users who try to do things beyond their normal capability, yet virtually no difficulties are put in place when they meet the appropriate criteria.

Of course this approach (as do other password approaches) relies on the users to keep their passwords to themselves and not tell other people. This approach provides Information Services with the ability to change passwords randomly thereby increasing the security of each application. However, certain software tools are required to do this properly. With this approach 40 man-hours would not be required to change the passwords in each of the applications.

SYSTEM DEVELOPMENT

When a system is being developed by Information Services personnel, the programmer is assigned the user name PROG and an associated password. All development takes place in the DEV group of the user's account. Programs to be modified are copied to the DEV group as would data files, etc. Development takes place essentially without bother to normal users.

Once development or modifications have been completed, they should be tested by the users. This could, and should, take place within the DEV group. This is a situational call by the staff of Information Services. Once this stage is completed and the user has "signed-off" all programs, data files, run files, etc. these files will be transferred to the appropriate locations. The password for PROG will be changed, thereby providing the user with assurance that the account will not be modified without their knowledge and permission.

MAINTENANCE OF SOURCE CODE

All source code for each account resides in the group SOURCE. However, source code will not always reside on disk. This would tie up a key resource unnecessarily. Production source code is maintained on two tapes for each account. One tape is kept in the computer room, the other in the vault. The purpose of this is to ensure that parity errors that could occur with tapes would be minimized. Should one tape be unreadable, the second is available as a backup. There is a very low probability that both tapes would be unreadable.

Should a program need to be modified, the following sequence would take place. The program(s) would be:

- * RESTORED to SOURCE.ACCOUNT

NAMING CONVENTIONS

Generally the following convention applies to all file names whether they are program, data or run files. The exceptions such as databases, some Quasar products and library routines take the same basic approach.

FILENAME = SSAAnnTv.GROUP.ACCOUNT where:

- | | |
|-------------------|--|
| SS = System: | The system or application alpha name, eg. AR for Accounts Receivable. |
| AA = Application: | The number assigned to the application or function or program series. This will start with "00". |

- * Copied from SOURCE.ACCOUNT to DEV.ACCOUNT
- * Renamed in SOURCE.ACCOUNT to version "1".
- * Modified, tested and "signed-off"
- * Copied from DEV.ACCOUNT to SOURCE.ACCOUNT as version "0".
- * Purged from DEV.ACCOUNT

If this is a volatile account, ie. frequent changes are being made to source code, transferring to tape would only be done periodically. Whether the account is volatile or not doesn't matter, the process of off-loading source code to tape is the same.

Time is the only point of concern for volatile accounts. The process of off-loading source is as follows:

- * STORE @SOURCE.ACCOUNT to a scratch tape (precautionary step only)
- * RESTORE @SOURCE.ACCOUNT with KEEP option
- * STORE @SOURCE.ACCOUNT to computer room tape
- * STORE @SOURCE.ACCOUNT to vault tape
- * Sign-on to the SOURCE group
- * PURGE the group SOURCE

This last step will purge all files within the group without actually purging the group itself. If, however, the group is purged without being signed-on to the group, the group as well as all the files within the group will be purged.

nn = Number: The sequential number applied to the application series, file, etc. This will start with "00".

T = Type: The type of file this represents.

V = Version: As the name implies, this is for version control. "0" is the current version - "1" indicates the most recent past edition - "2" the next most recent past edition, etc. We would normally not keep more than version 1 for archiving purposes. This policy, however, is determined by circumstances.

It is the responsibility of the Database Manager to assign, maintain and control file names to ensure integrity. The system and application names ie. "SS" and "AA" of the filename, are assigned by the Database Manager. To some extent, programmers will find it necessary to have license over the "AA" portion of the name. When this is the case, the initial "AA" is assigned by the Database Manager. Discretion must be used by the system designers when this control is required. The Database Manager is informed of all changes.

FILE TYPES

The type of files being identified are as follows:

T	FILE TYPE
-	-----
S	COBOL source code
E	SPL source code
A	FORTTRAN source code
C	BASIC source code
V	VIEW source code
F	VIEW fast forms file
Z	QUIZ source code
Y	QUIZ compiled code
K	QUICK source code
X	QUICK compiled code
G	QTP source code
H	QTP compiled code
D	Sequential data file
J	Job Streams
U	USL files (also see USE files)
P	Segmented executable code

Exceptions to the above rules are outlined below.

DATABASES

IMAGE databases are defined as follows:

SCHEMA:	SSAA.SOURCE.DATABASE
ROOT FILE:	SSAA00.IMG.ACCOUNT
DATA SETS:	SSAA00nn.IMG.ACCOUNT

QUASAR DICTIONARIES:

SOURCE:	SSQSAASv.SOURCE.DATABASE
COMPILED:	SSQSAADv.RUN.DATABASE

SS = System name
AA = Application name
S = Source code
D = Data code
nn = Sequential number assigned
by IMAGE
v = Version
QS = Quasar product

USE FILES

Use files used by system designers are maintained in the LIBRARY account in the group USE. The nomenclature for use files is:

ISPPnnUv.USE.LIBRARY where:

IS: Information Services
PP: Program type
nn: Sequential number
U: File type representing Use
v: Version

Program types are defined as follows:

ED: EDITOR
QK: QUICK
QZ: QUIZ

QT: QTP

COPYLIBS

The nomenclature for copylibs follow the standards outlined below:

SYSTEM.COPY.LIBRARY

OR

SSCPYLIB.SOURCE.ACCOUNT

The file SYSTEM.COPY.LIBRARY contains all system related copylibs while SSCPYLIB contains all copylibs belonging to SS. Each module is named for convenience such as INVMST for Inventory Master. The great majority of the copylibs that are used to define record layouts are defined by the Database Manager by default in the design of the database and the Quasar schemas.

John D. Baker, P.Eng., is Director of Information Services at Kingston General Hospital located in Kingston, Ontario, Canada. His twenty years of professional experience include management consulting with big eight accounting firm, managing the systems engineering function at a large metropolitan hospital and work as a systems analyst and staff engineer. Mr. Baker has taught college level courses and is the author of numerous articles in professional journals. He holds degrees in both industrial and mechanical engineering and is a member of several engineering professional societies.
