

**Introduction to the
INFOS™ System
with the
RDOS/INFOS™
System
Documentation Guide**

093-000113-02

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 093-000113
©Data General Corporation, 1975,1977
All Rights Reserved
Printed in the United States of America
Revision 02, November 1977
Licensed Material - Property of Data General Corporation

NOTICE

The information contained in this manual is the property of Data General Corporation (DGC) and shall not be reproduced in whole or in part without DGC prior written approval.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

Introduction to the INFOS™ System
with the
RDOS/INFOS™ System Documentation Guide
093-000113

Revision History:

Original Release - February 1975
First Revision - February 1975
Second Revision - November 1977

This document has been extensively revised from revision 01; therefore, change indicators have not been used.

The following are trademarks of Data General Corporation, Westboro, Massachusetts:

<u>U.S. Registered Trademarks</u>			<u>Trademarks</u>
CONTOUR I	NOVA	NOVALITE	DASHER
DATAPREP	NOVADISC	SUPERNOVA	INFOS
ECLIPSE			microNOVA

Introduction to the INFOS™ System



Today we are talking with Stan Engles, Data General's resident expert on the INFOS system. Stan has agreed to tell us a little about the INFOS system and its applications.

Q: Good afternoon, Stan. The first thing we'd like to know is: What is the INFOS system?

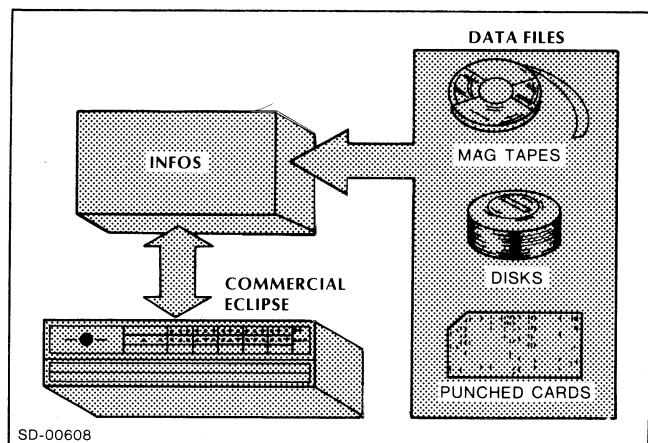
Stan: Well, to put it simply, it's a computer program that organizes and manages files, and it runs on a Data General Commercial ECLIPSE computer.

Q: What can we do with it?

Stan: You can create, use, and maintain simple or complex data files of almost any kind of information. Programmers can organize the data in many different ways, and Systems Analysts can fine-tune the INFOS system for peak performance in a particular application.

Q: Are you saying that INFOS is the perfect system?

Stan: No, it's not perfect, but since it is a flexible, general-purpose information processing system, you can successfully apply it to just about any application.



Q: Can you give us a couple of examples?

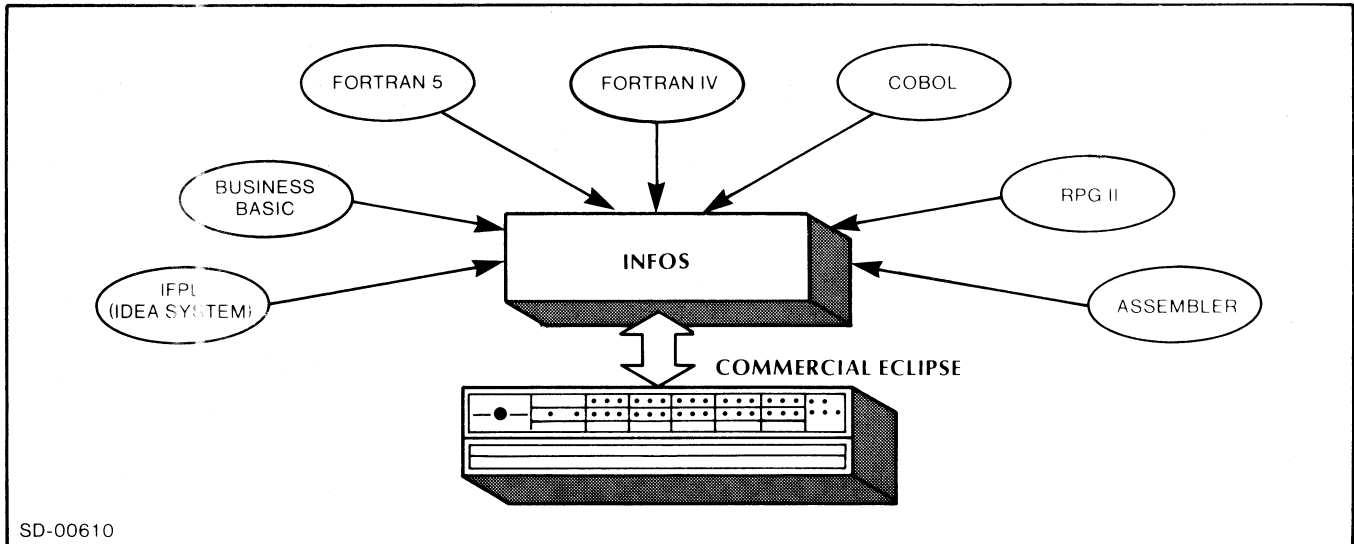
Stan: Sure. You could use it to keep track of, and issue reports on, customer accounts by name, account number, payments per month, items ordered, or whatever you want. Also, it's especially useful if you have to work with a lot of frequently-changing data, such as an inventory file whose records change each time you buy or sell parts - or a hotel billing system where data flows in and out as guests arrive, depart, and use room service or the telephone.

INVENTORY CONTROL	
ITEM	STATUS
RAVIOLI, FIRST CLUB.	16 OZ.
IN STOCK:	56 CANS
SOLD TODAY:	5 CANS
PRICE:	\$.59
ON ORDER:	0 CASES

SD-00609

Q: Hmm. Can my programmers use COBOL with INFOS?

Stan: Definitely. In fact, they can code their programs in COBOL, RPG II, FORTRAN IV or 5, Business BASIC, or IFPL (the IDEA system's language) or they can use our Macroassembler.



SD-00610

Q: That's very handy. What are some of the system's general features?

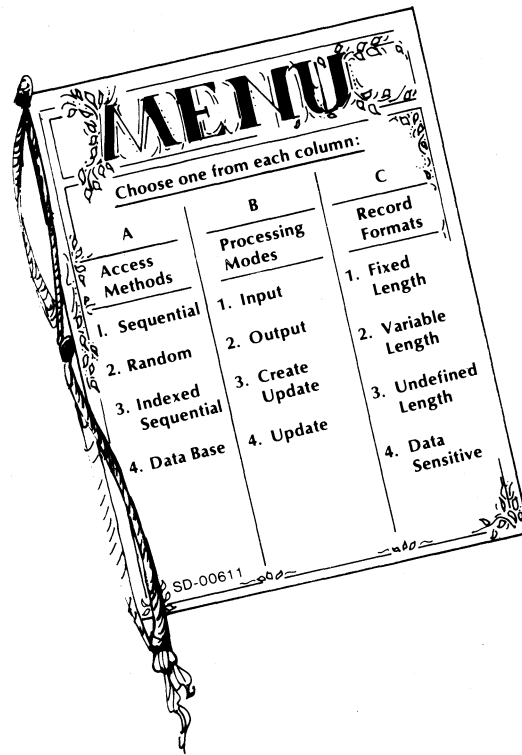
Stan: Well, let's see... we'd have to include:

- File Sharing, which lets several users at separate terminals use one INFOS file simultaneously;
- Space Management, which helps you eliminate wasted time and storage space by automatically reorganizing your file each time you delete or update a record;
- Lockout Options, which you can use to prevent other users from accessing either the file or the record you're using;
- Character Code Translation, so that you can use files generated on another system (IBM, for instance) as part of your INFOS system;
- Labeled Magnetic Tape Processing, so that you can use any industry-standard labeled tapes with the INFOS system; and
- The INFOS Utilities, which, among other things, let you easily create, delete, sort, copy, rename, and examine your files without writing a specific program.

Q: Not bad, but how about some details?

Stan: Of course. In the INFOS system, good things seem to come in fours. To start with, you can choose from four different ways of getting to your data, or *access methods*. Then, once you've selected one of these, you can use any of four *processing modes*, depending on what you want to do with your file. If you just want to read the file, you'll use the Input mode; if you only want to write to your file, choose Output; when you want to read *and* write, you can use Update or Create Update.

The final group of four contains the *record formats*. Depending on the access method you choose, your records can have fixed, variable, or undefined lengths, or, by selecting the data sensitive format, you can specify a character as an end-of-record indicator.

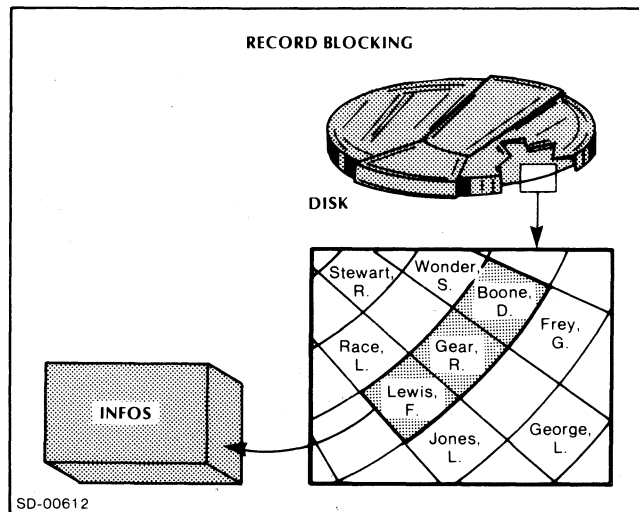


Q: So I can tell the system how I want to *access* my records, what *format* they're in, and how I want to *process* them.

Stan: Exactly.

Q: But won't all these details increase my processing time?

Stan: No, not at all. In fact, the INFOS system significantly *reduces* your processing time by grouping data records into sets, called blocks. That is, when the system initiates a data transfer, it takes a whole set of records at once, instead of transferring them one at a time. This block method is consistent throughout the system, and helps you specify and maintain large amounts of data with very few physical restrictions.

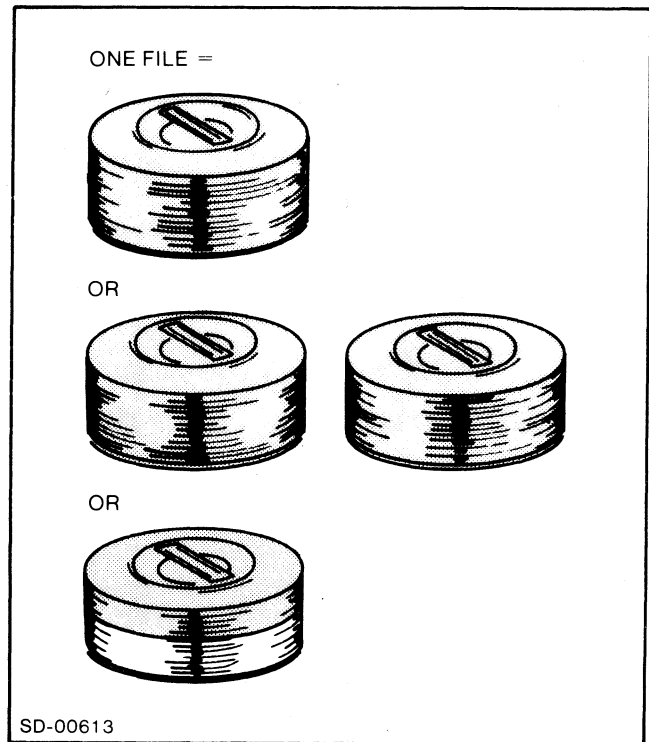


Q: But my files will be quite large. Can I store them on more than one tape or disk?

Stan: Yes. Your INFOS file can occupy as much or as little space as you need. If you use disks, your file can reside on part of a disk, or on one entire disk, or on many disks. And the same goes for magnetic tapes. As I said, flexibility is one of the INFOS system's strong points.

Q: Earlier, you mentioned that the system provides four ways to access data. Why are there so many?

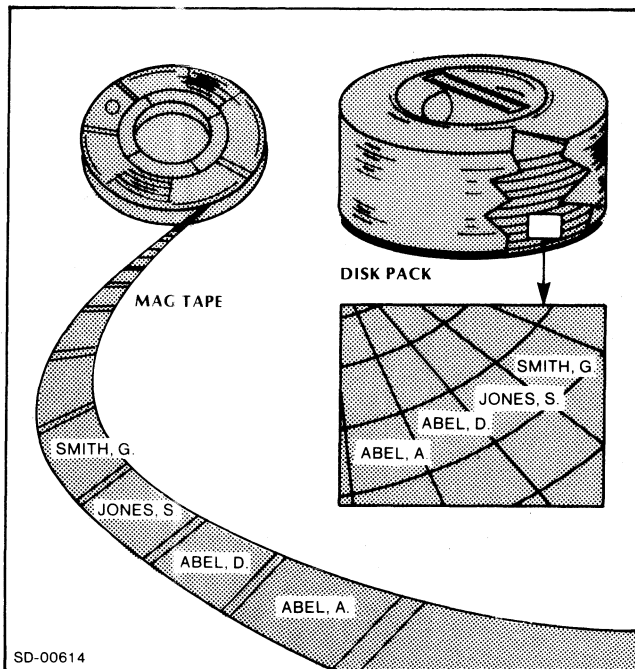
Stan: Simply to make it as versatile as possible. Every user of the system will have different applications for it; by providing four different access methods, each user can tailor the system to fit each application, from simple sequential files to complex data structures.



Q: What are the differences among the four methods?

Stan: Well, let's start with the Sequential Access Method, or SAM.

SAM provides the fastest means for processing your data, *if* you want to process it sequentially. That is, when you use SAM, your data is processed in the order in which it appears in your file. In addition, the system automatically anticipates your next request for data, and gets that data while it is working on your current request. It also allows you to write to your file without waiting.



Now, this is fine if you're processing a number of records at one time, but it may not be best if you want to access only a few records. For example, if you want to read record number 100, you have to go through records 1 through 99 to get to it. Also, adding data to, or removing it from a SAM file usually means that you have to copy the entire file, which can take a lot of time. However, as I said, this *is* the fastest access method, *if* you want to process your data sequentially.

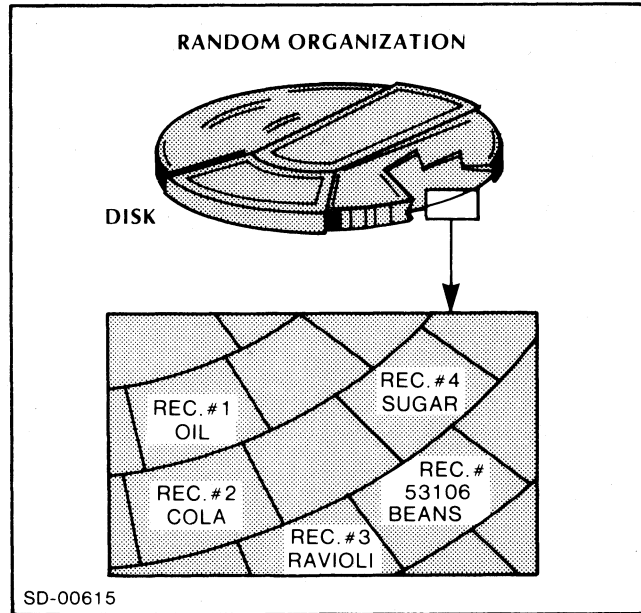
Q: What if I want to read record 100 *without* reading records one through 99?

Stan: Well, there's always the Random Access Method, or RAM.

RAM lets you access your records in any order, regardless of where on your disk(s) they're physically located. That is, unlike SAM, you don't have to go through a lot of records to get to the one(s) you want. You just give the system the number of the record you want, and it retrieves it for you.

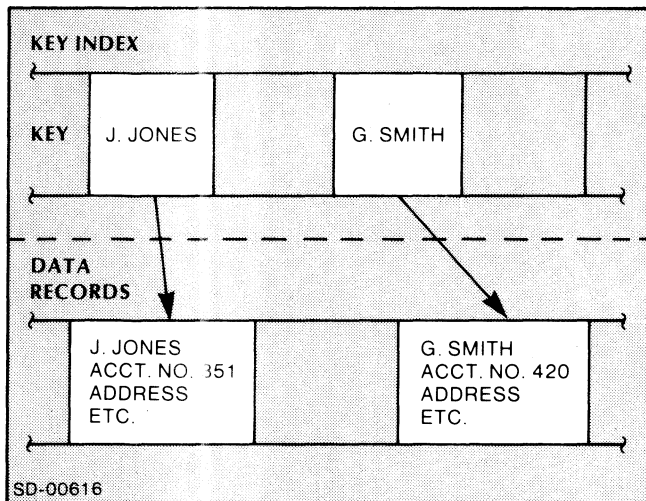
However, a record number will often have no logical connection to the record, so you have to establish your own relationship between the record and the record number. In other words, you may find it difficult to remember that, to the system, a can of beans is a 53106.

Then again, you can access your records very quickly, as you need them, and your INFOS RAM file can hold up to 4-1/4 billion records.



Q: That's fine, but a lot of different people in my firm would be using my INFOS system, and it's going to take a lot of time for them to find a record number if they don't remember it. Can't we avoid this problem?

Stan: Sure, with our third access method, the Indexed Sequential Access Method, or ISAM. When you have a complex data file which you want to access quickly through a number of terminals, it's much easier to retrieve data using something which is part of the record you want, such as the customer name, or an account number, or a part name. We call these things *keys*.



Q: But doesn't RAM use record numbers as keys?

Stan: Well, sort of, but those record numbers usually aren't part of the data records themselves, and they can only be numbers. ISAM's keys are different because they usually are part of the record (although they need not be), and they can be letters as well as numbers. When you want to read John Jones' record, you just say READ "JONES,J." This is known as *keyed access*.

Q: How long can these keys be?

Stan: As long as you need, up to 255 characters.

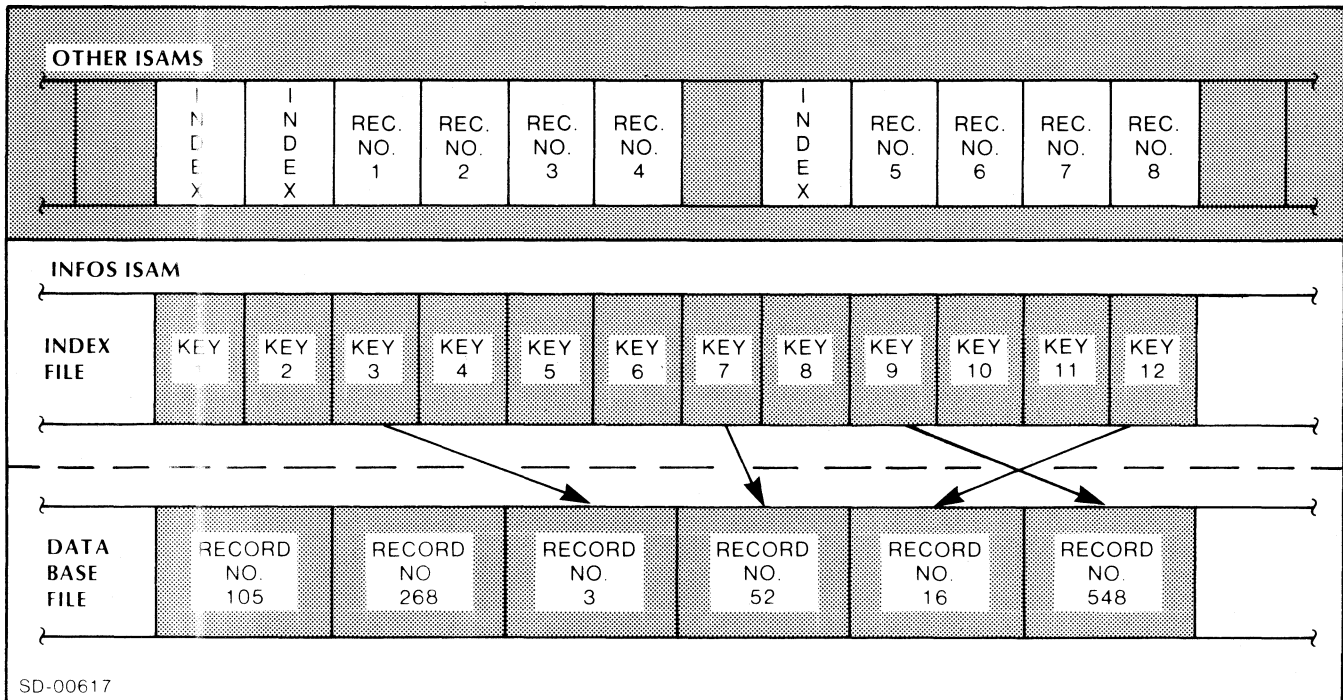
Q: Do they all have to be the same length?

Stan: Not at all. If you want to call one record "Petty Cash" and another "Accounts Receivable for March, 1976" it's okay.

Q: A few years ago we tried to use an ISAM file, but it was so time-consuming and used so much storage space that we gave up on it. How can the INFOS ISAM be any different?

Stan: Other systems do offer ISAMs, but they all have a major flaw: they're only one file. That is, the indexes which hold the keys are usually interspersed among the records so that the keys point only to a group of records. The system still has to go through each record in that group to find the one you want. As you found out, this takes extra time, and it can very inefficiently use the storage space.

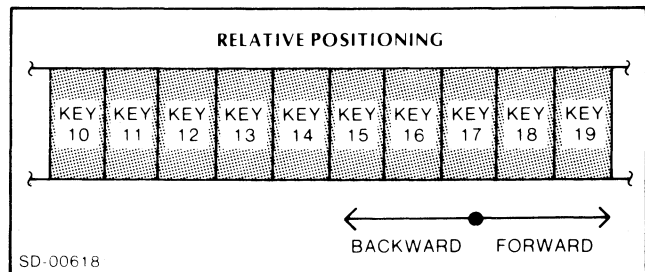
The INFOS ISAM, however, is actually *two* files: one for the index, and one for the data base records themselves. The INFOS system automatically maintains the relationship between the index and the data base file, so that each key points to its data base record. This saves processing time as well as storage space.



Q: Keys sound very efficient for random access in an ISAM file, but what if I want to process my file sequentially?

Stan: No problem. In addition to keyed access for random processing, the INFOS system also lets you use relative position processing.

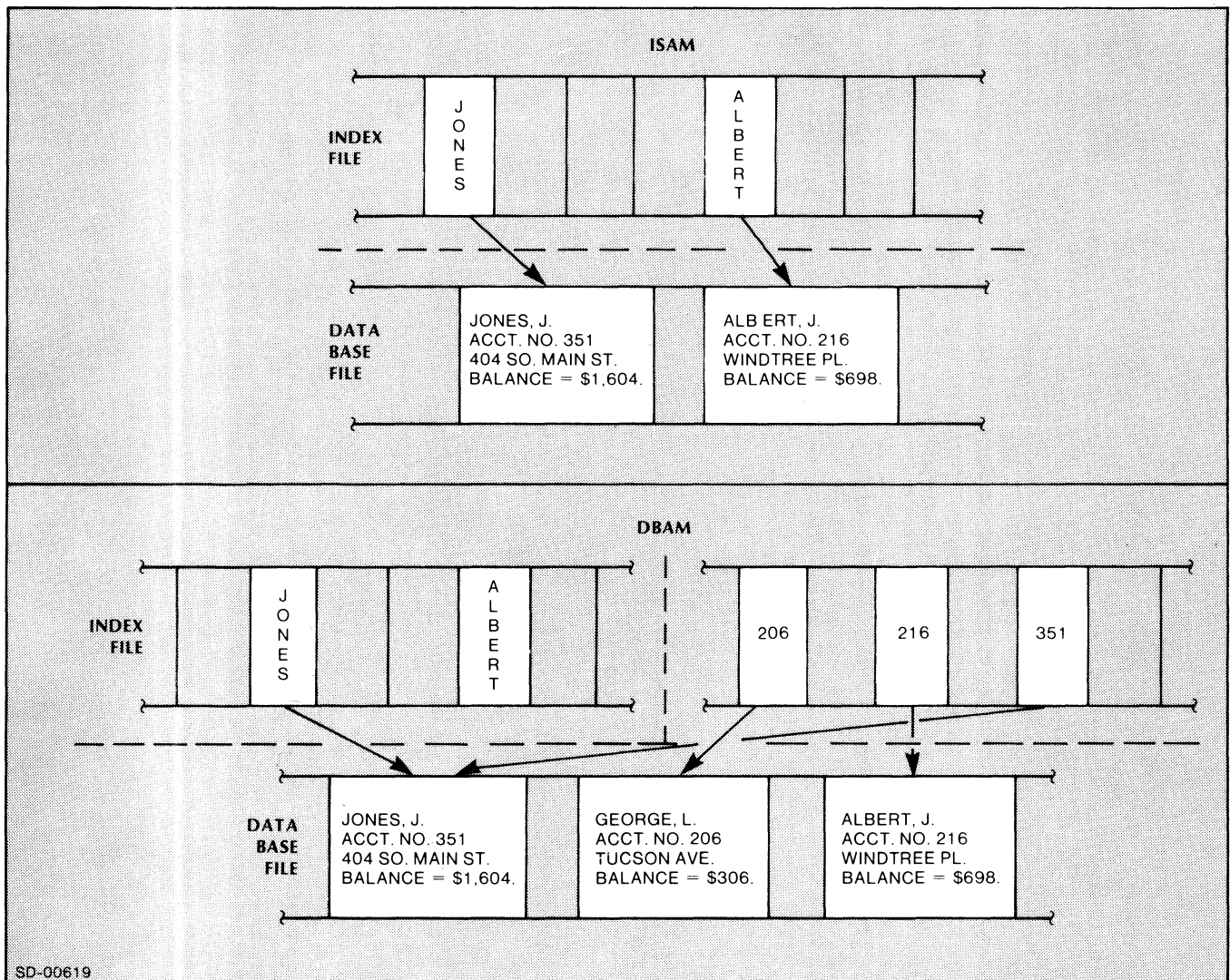
This allows you to move sequentially through your ISAM file in the order of your index keys. We call it relative motion because all movement within the index is done relative to your current position. If you want to read a record which you have already passed by, you move backward; if you want to read the next record, you move forward. In fact, you can even use keyed access to get into your file, then use relative positioning to move sequentially through your file from that point.



Q: Okay. INFOS ISAM does sound a lot better than the other one we tried, but we're going to want to do some fairly complex operations with our file. As flexible as your ISAM is, it doesn't sound like it can cover all of our needs. For instance, what if we know only a customer's account number, but not his or her name? Under ISAM, we're stuck; we can't get to the customer's record.

Stan: Well, perhaps you should consider our fourth access method -- the Data Base Access Method, or DBAM. DBAM is like a super-ISAM; it extends all the capabilities of our ISAM so that you can design very complex data structures, yet use them easily.

For example, say you have a file in which you keep customer information -- name, account number, address, and so forth. Under ISAM, you're right, you can only access that file with one set of keys; for instance, the customer name. Under DBAM, however, you can build another index based on the account number, or the address, or the balance, or whatever you like, to access the same information.



Q: But wouldn't you have to duplicate all the data base records?

Stan: No, and that's exactly my point. Under ISAM, you have only one index for one data base; under DBAM, you still have only one data base, but you can access it by as many different indexes as you want. This is called data base inversion.

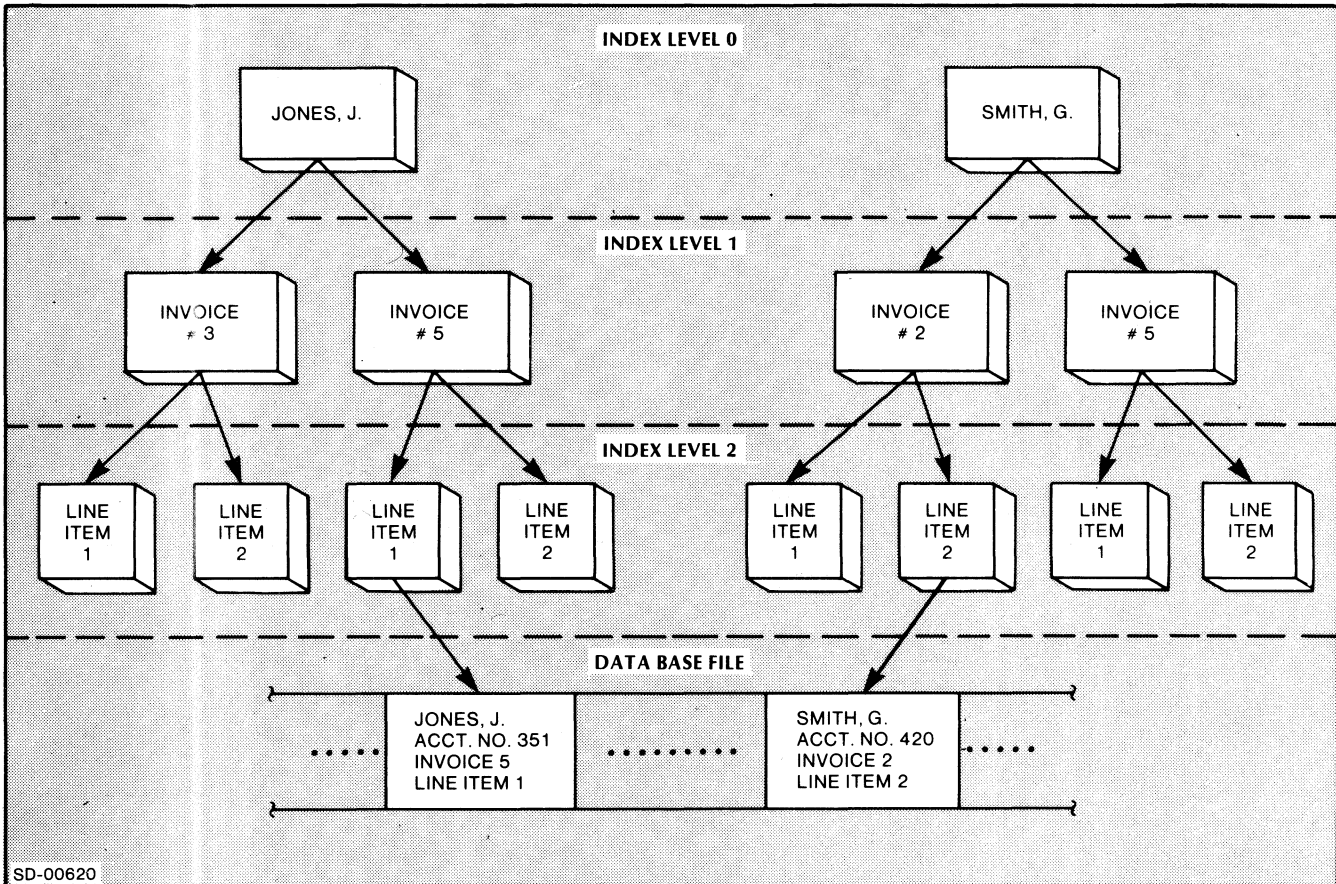
Q: That sounds like it needs a lot of storage space.

Stan: Not at all. Since keys usually don't take up much space in the index file, and since you have only one data base file, you need very little extra storage space. However, this is only one of DBAM's features. Another is multilevel indexing. In ISAM, you give the system one key and it gives you one record. In DBAM, you can give the system a series of keys which will lead you to a portion of that record.

Q: What do you mean?

Stan: Let me give you an example.

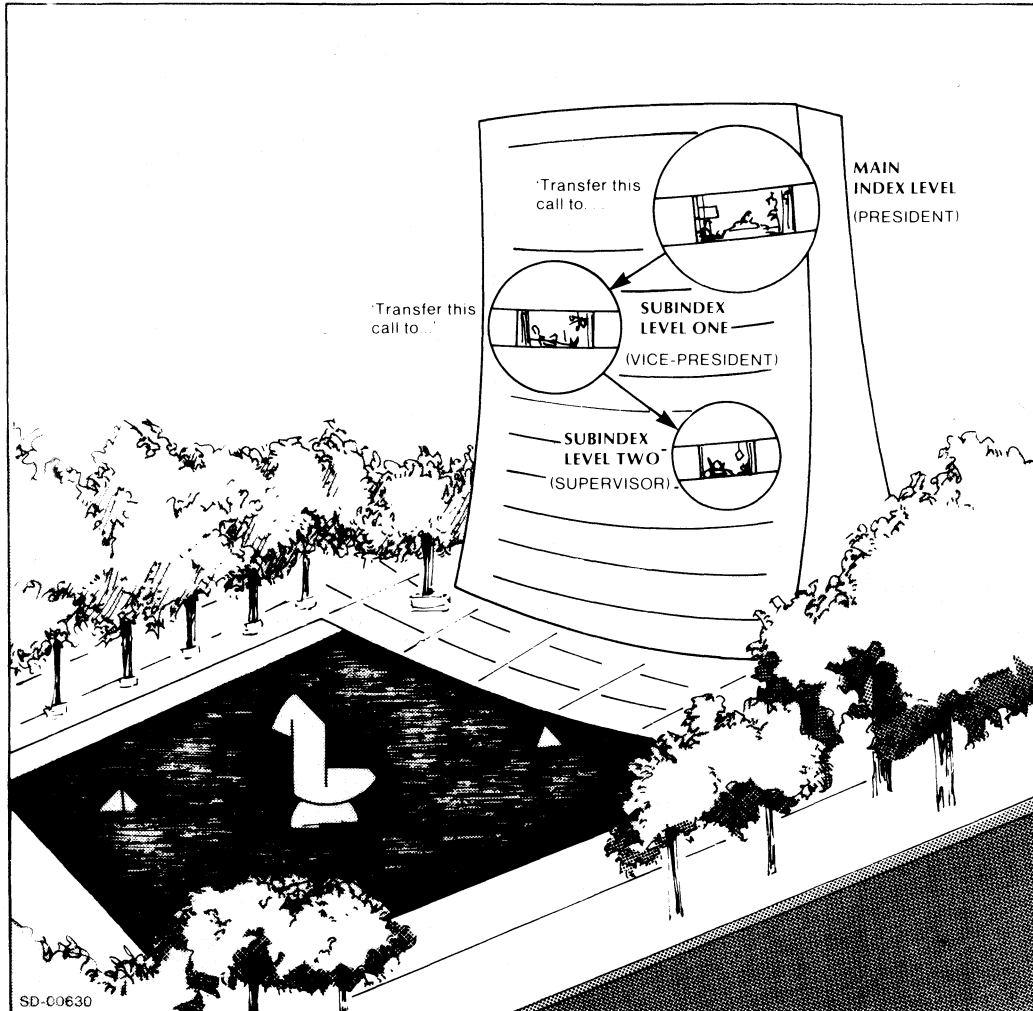
In ISAM, you would say READ "JONES,J" and you would get John Jones' record. Then you would have to go through his entire record to get the information you wanted. If his record held information about a number of invoices, it would take you some time to get to the one you wanted. In DBAM, however, you could say READ "JONES,J", "INVOICE 5", "LINE 2" and the system would tell you the second item on invoice number 5 for John Jones.



SD-00620

Q: How do I get the system to do that?

Stan: By setting up different *levels* within your index. That is, "JONES,J" would be on the main index level, "INVOICE 5" would be on the first level below that one (or, in the first *subindex*), and "LINE 2" would be on the second subindex level. In other words, you're telling the system that you want it to find the whole record ("JONES,J"), then find a particular section of that record ("INVOICE 5"), then read a part of that section ("LINE 2"). It's a little like calling the president of a large company with a complaint: He'll transfer you to the vice president in charge of the responsible division, who will transfer you to the supervisor of the department which made the mistake.



Q: Can I combine multilevel indexing with the inversion feature you just explained?

Stan: Of course.

Q: Then I could say READ "ACCOUNT 351", "INVOICE 5", "LINE 2" and get the same information?

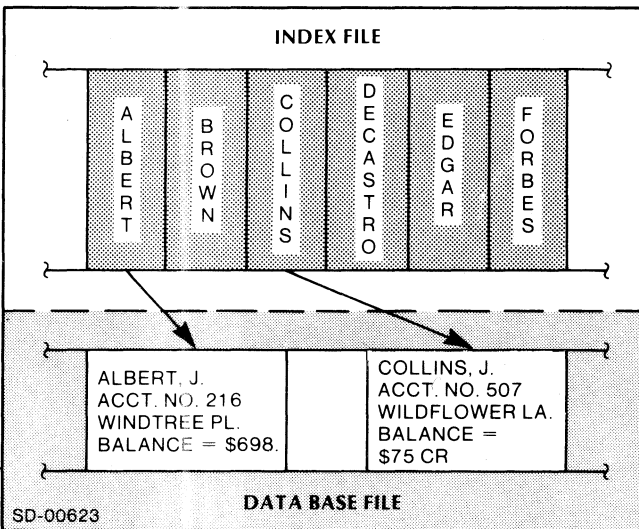
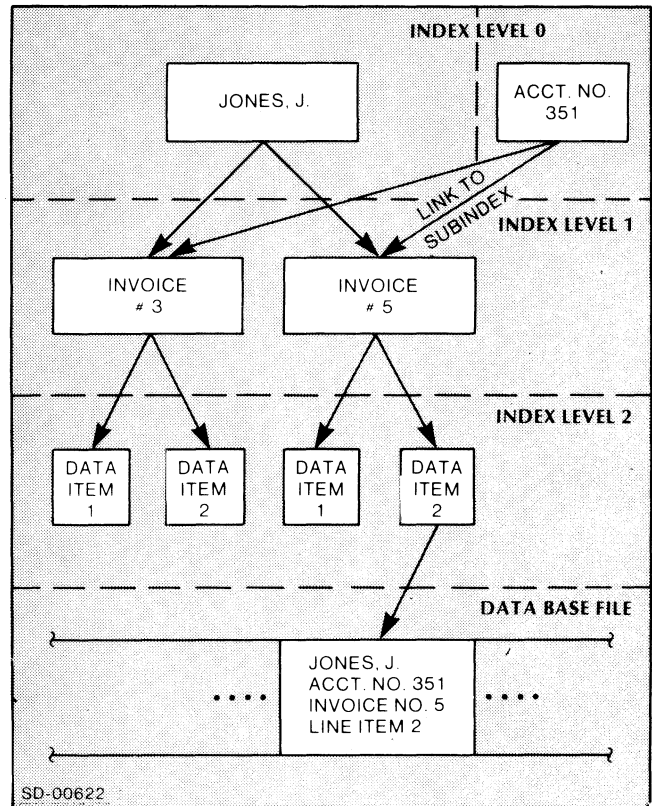
Stan: Assuming that John Jones' account number is 351, and that you have set up your indexes properly yes. That's known as sharing subindexes. That is, your account number index could point to the same set of subindexes as your customer name index.

Q: But it doesn't have to.

Stan: Right.

Q: What else can I do with DBAM?

Stan: Well, while we're talking about keys, there are several "key" features, so to speak. For example we've been assuming that your keys are part of their data base records, but DBAM keys don't *have* to be contained in the record.



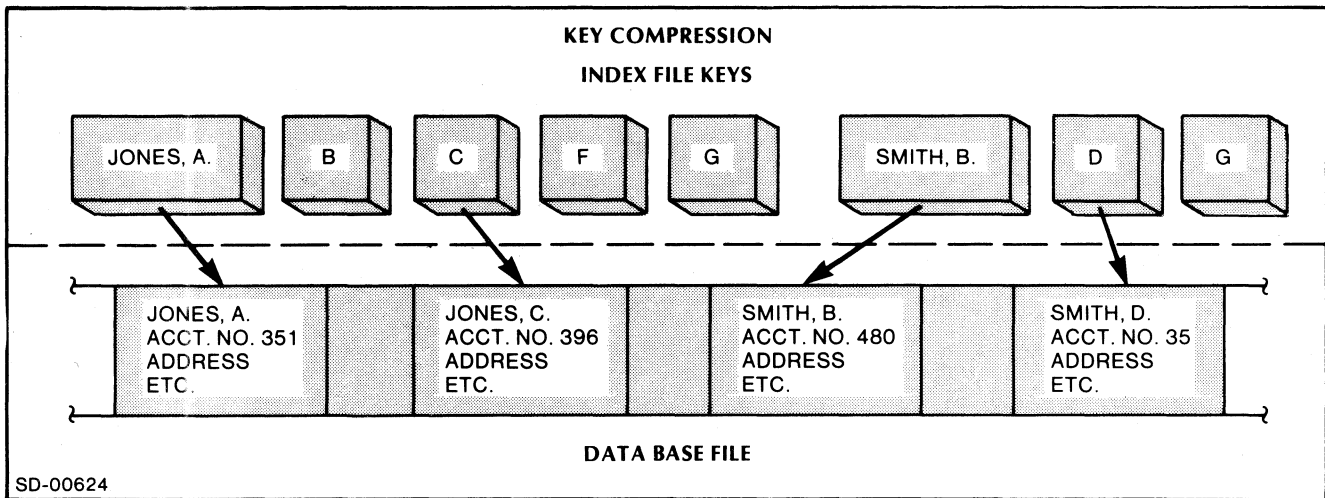
You can call a record by any key you want when you build your index file. In fact, you can even set up a key which is not associated with any record.

Q: You mean that if I knew that I was going to be getting data for my file, I could just set up its key now and fill in the data record later?

Stan: Right. Now tell me something: Do you have several customers who share the same last name?

Q: Yes. Why do you ask?

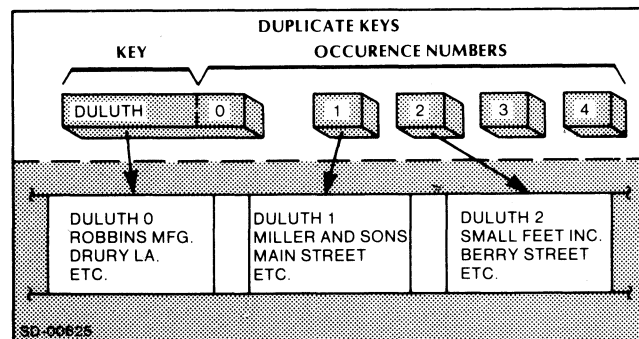
Stan: Because DBAM also has a feature called Key Compression which saves you storage space in your index.



If you have several keys with identical information, Key Compression will tell the system to store only the unique part of each key. For instance, if you're using customer names as keys, you may find that several customers have the same last name, say Smith, as their record keys. The INFOS system can store the common part of these keys just once in the index, and the unique part of each key (perhaps the first initial) in each index entry. Of course, to read any of those records, you have to specify the whole key ("SMITH, G"), but the system isn't wasting space storing all the duplicate characters.

Q: What if I want to use one key to refer to several data base records?

Stan: That's easy; we call it Duplicate Keys. If you request this feature, the system will automatically assign an occurrence number to each duplicate key. That way, you can get to any of the records by stating a key and an occurrence number. For example, if you work on a regional basis, you may want to group all of your accounts in the Duluth area under the key "DULUTH". If you specify Duplicate Keys, the first Duluth account will have the key "DULUTH 0"; the second will be known to the system as "DULUTH 1", and so forth.

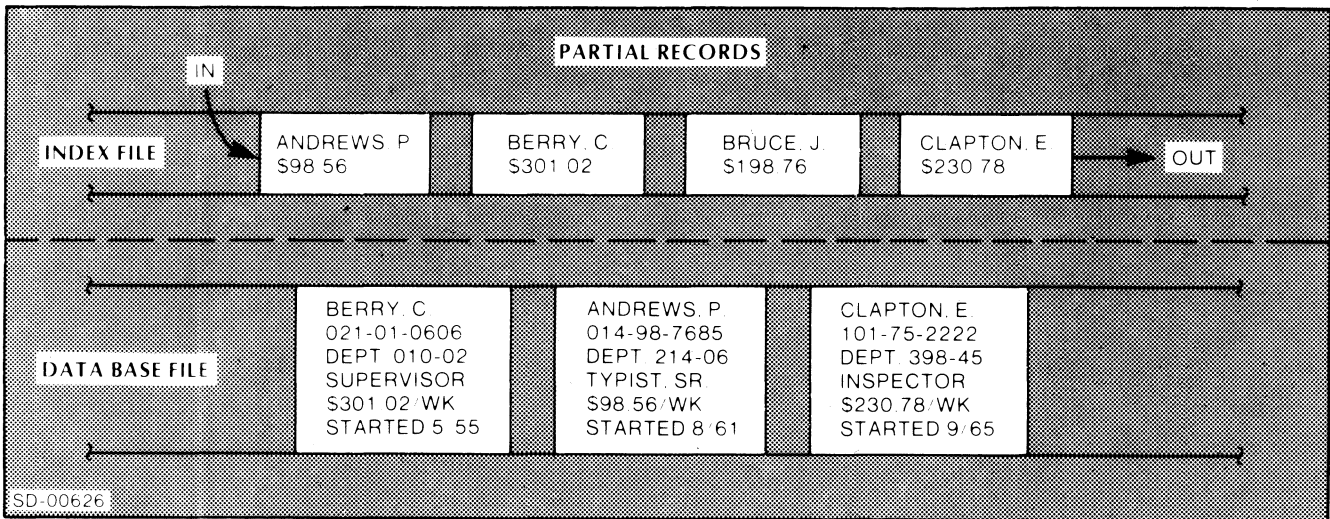


Q: That's pretty handy.

Stan: Almost as handy as Partial Records.

Q: Huh?

Stan: Well, say you have a personnel file which contains, among other things, employees' salary information which you use each week to generate paychecks. You'd waste an awful lot of time if you had to go through each employee's record just to retrieve that one piece of information, so the INFOS system allows you to store that data in your index file along with the keys. That way, you only need to access the keys; you never have to go to the employees' data base records.

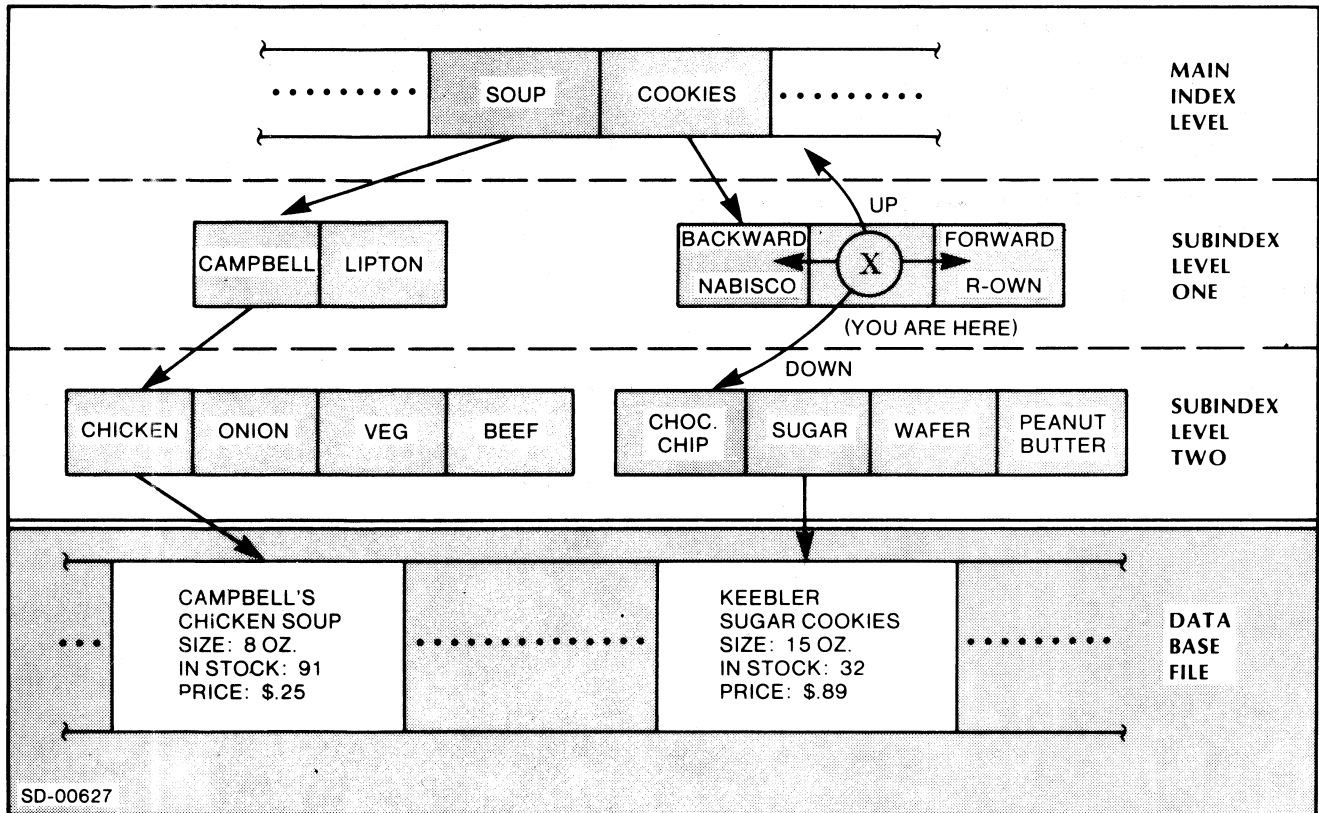


Q: That does sound useful. But what if I don't know, or forget, part of a key? Can I still access the records I need?

Stan: Sure. The INFOS system's Generic and Approximate Keys will lead you to the records, even if you aren't sure of their keys. You'd use a Generic key if you know only part of a key, and an Approximate one if you think that the key is "something like (x)".

Q: That's very interesting, but in order to get the reports we want, we're also going to want to process our file sequentially. Does DBAM have anything which corresponds to ISAM's relative access?

Stan: Indeed it does. I mentioned earlier that you can use subindex levels in DBAM, and we've enhanced the relative access capabilities of ISAM to make working with this structure easier. For example, in addition to forward and backward motion, you can also move up and down to different subindex levels. Also, you can *combine* keyed and relative access into one positioning request to get you exactly where you want to go within your index.

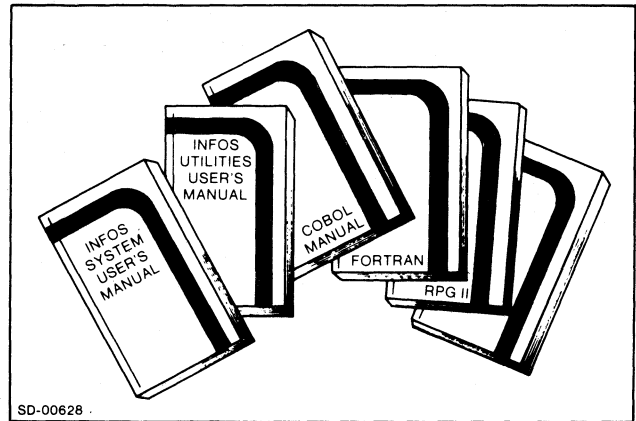


Q: That sounds confusing.

Stan: It's not, really, once you get the hang of it. And I guess that's true of the whole INFOS system. But the nice thing about the system is that it's fully documented. The *INFOS System User's Manual* gives you all the details about the points I've only been able to touch on today, including how to program the system to fit your individual application. Plus, as I mentioned earlier, the *INFOS Utilities User's Manual* comes in very handy once your system is up and running. And, of course, you can refer to the *Documentation Guide* to find out more about those manuals you should read.

Q: Thank you, Stan, for taking the time for this introduction to the INFOS system.

Stan: It's been my pleasure.



End of Interview

RDOS/INFOS System Documentation Guide

As you discovered in the interview, the INFOS system is a sophisticated and flexible processing system which you can use in any number of applications. In addition, it uses Data General's Real Time Disk Operating System (RDOS), so you will probably have to read more than one manual in order to fully understand how to make the INFOS system perform in your application.

Depending on your background and interests, you will find certain manuals more applicable than others. For example, if you are a COBOL programmer, you don't need to (and probably won't want to) read about the inner workings of our Macroassembler, but you will want to read about DG's COBOL, as well as about the INFOS system itself.

The table below is your guide to the manuals which are most applicable to you. To use the table, just pick the position description which most closely describes your job and the language(s) you will be using. The letter series at the junction refers to the manuals described below. Furthermore, you should read the manuals in the order they are listed in the table. In other words, if you're an RPG II programmer, you should read manual A first, then manual L, then manual C.

Note that most of our manuals are written in a modular format, so you probably won't have to read all the chapters in every manual listed.

A description of each letter series follows. Use the part number following each title to order the manuals you want.

Series Manual

A *Introduction to INFOS*, 093-000113 (i.e., this booklet). This manual highlights the RDOS/INFOS system's features and facilities. It is easy to read and does not go into detail.

B *The INFOS Storybook*, 093-000199. This is the story of how a fictitious (yet plausible) company successfully applied the INFOS system to its data processing problems. The book shows how the INFOS system can be used in a real-world situation.

C *The RDOS/INFOS System User's Manual*, 093-000114. This manual is the primary reference for anyone who wants to fully understand and use the INFOS system. It provides all the details behind the INFOS system: what it does, how it does it, and how you can make it fit your application. Its Planning section describes the system in plain English, and its Programming section tells programmers how to code applications in both FORTRAN and assembly language.

Position	COBOL	RPG II	FORTRAN	Assembly
System Operations Manager	A, (B), C, E, S, G, (F), I, (P)	A, (B), C, L, E, G, U, (I, P, F)	A, (B), C, E, Q, S, G, I, F	A, (B), C, E, U, S, G, D, P, F
Senior Programmer/Analyst	A, (B), N, M, C, G, O	A, (B), L, C, G, K	A, (B), Q, R, C, G	A, (B), C, T, D, H, K, G, (P)
Programmer/Analyst	A, (B), N, M, C, O	A, (B), L, C	A, (B), Q, R, C	A, (B), J, T, D, H, C, K, I

() = Optional

- D *Extended Assembler User's Manual*, 093-000040. This manual provides the details of Data General's Extended Assembler. It details all the assembler functions, as well as providing information on writing source programs, using atoms, expressions, the ECLIPSE instruction set, and useful pseudo-ops. This manual is a must if you're going to code your INFOS system application in assembly language.
- E *The INFOS Utilities User's Manual*, 093-000182. The INFOS system provides several utility programs which allow you to create, delete, copy, rename, and inquire about your files. This manual tells you how to use each of these utilities, and includes the full texts of all interactive dialogues, along with possible responses.
- F *ECLIPSE-Line Sort and Merge Utilities Programs*, 093-000126. The Data General Sort utility is a general purpose sorting program that orders files of data records according to a predetermined collating sequence. This manual describes how to use these utilities, and includes information on sort control statements, user code, executing the sort, merge control statements, etc.
- G *Real Time Disk Operating System (RDOS) Reference Manual*, 093-000075. This is your primary reference manual if you want to gain a comprehensive understanding of RDOS. It describes the features of both the mapped and unmapped versions of RDOS for ECLIPSE computers, including files, directories, disk partitions, system communication, using your address space, multitasking, and foreground/background programming.
- H *ECLIPSE Macroassembler*, 093-000081. This manual describes the rules and conventions you must follow when using the Macroassembler, and it covers topics such as syntax, atoms, permanent symbols, and the macro facility.
- I *RDOS User's Handbook*, 093-000105. We designed this handbook as a console reference to remind users of the features of RDOS and its subsidiary systems. It contains CLI command descriptions, subsystem commands, and the meanings of RDOS error messages.
- J *Fundamentals of Small Computer Programming*, 093-000090. If you've never programmed in assembly language before, or if you have never used a small computer, you should read this manual. It covers programming techniques, and discusses the following: elementary computer hardware concepts, binary arithmetic, flowcharting, and elementary programming considerations.
- K *RDOS CLI Reference Manual*, 093-000109. The Command Line Interpreter is the primary interface between you (at a system console) and the INFOS system. You can use it to start utilities, execute programs, perform simple file maintenance, etc. This manual describes all the features, commands, and operating considerations of the CLI.
- L *RPG II Programmer's Reference Manual*, 093-000117. If you're going to code your INFOS application in RPG II, you must read this manual. It describes Data General's RPG II in detail, including all specifications - control card, file description, extension, line counter, data, input, output, etc.
- M *ECLIPSE COBOL Programmer's Reference Manual*, 093-000180. A must-read for COBOL users, this manual describes in detail Data General's COBOL and the features for compiling, executing, and debugging your COBOL programs.
- N *Introduction to COBOL*, 093-000181. This book will start you programming in Data General's COBOL. It explains the fundamental features of COBOL, but does not spend time on less common details. Novice programmers (or those merely unfamiliar with COBOL) can use this manual to learn the language and write simple COBOL programs.
- O *COBOL Pocket Reference*, 093-000200. This handbook summarizes all the information contained in the *COBOL Programmer's Reference*, but it does not explain how to use the various commands, etc.
- P *RDOS User Device Driver Implementation*, 017-000018. This manual outlines the methods of adding user-assignable multipriority servicing routines to handle interrupts. You can include these device drivers as part of the operating system, or you can implement them as part of your application and attach them to an interrupt dispatch program.

Q *FORTRAN IV Reference Manual*, 093-000134, and *FORTRAN 5 User's Manual*, 093-000085. These manuals describe the language features and operation of Data General's FORTRAN IV and FORTRAN 5, including both ANSI-standard features and extensions.

FORTRAN 5 Supplement, 093-000185. This document supplies information on operating procedures, runtime environment, and multitasking in the FORTRAN 5 system. It also illustrates how to call assembly language subroutines using FORTRAN 5, and how to call FORTRAN 5 subroutines using assembly language.

FORTRAN Commercial Subroutine Package (CSP), 093-000107. This describes the CSP software package, which lets you handle business data efficiently in both FORTRAN IV and FORTRAN 5.

R *FORTRAN IV Runtime Library User's Manual*, 093-000142, and *FORTRAN 5 Runtime Library User's Manual*, 093-000096. These manuals describe the size, calling sequence, and other features of each routine in the FORTRAN IV and FORTRAN 5 runtime libraries. Each manual also explains how to interface assembly language routines to a FORTRAN program and the FORTRAN runtime stack mechanism.

S *Introduction to the Real Time Disk Operating System (RDOS)*, 093-000083. This manual presents a broad overview of the capabilities and features of RDOS, including organization, communication, task calls, etc.

T *Programmer's Reference Manual to the ECLIPSE C Series Computers*, 015-000047. This hardware manual tells you all about the features of the ECLIPSE computers, their internal structures, and their instruction set.

U *How to Load and Generate Your ECLIPSE-Line RDOS System*, 093-000189. This manual gives you step-by-step instructions for loading your RDOS/INFOS system into your computer. Also included is information on building a tailored INFOS system, saving and restoring INFOS, and initializing disks.

What About Changes and Errors?

Naturally, no manuals are perfect. Moreover, last-minute information can change what has already been printed. Instead of asking you to wait for new manuals to be printed (this could take a long time), Data General has two ways to get the information into your hands: Addenda and Release Notices. Addenda are replacement pages; they update and correct information in their parent manuals. As we print new revisions of manuals, we incorporate addenda information into the new manuals.

Release Notices are different from addenda. Data General provides a release notice with each major software product that is shipped to you. Customers enrolled in the Software Subscription Service receive subscribed software with new release notices whenever they become available. We print release notices on a line printer; thus they contain the latest programming information, which we could not provide in standard manuals. This information may include any or all of the following:

- software part numbers and model numbers;
- a description of file organization and layout on software tapes;
- a complete list of supplied documentation;
- a description of prerequisites to using the software;
- a list of the latest revisions of ancillary software used with the product;
- a description of enhancements in the latest software and warnings concerning its use;
- any known patches;
- errata information for the software documentation.

Function information in release notices is incorporated into later revisions of the manuals.

What Do the Manual Part Numbers Mean?

We employ the following conventions in our documentation part numbers. Each part number consists of three fields:

- an initial three-digit field for each general category of documentation;
- a six-digit specific part number;
- a trailing two-digit field showing the manual's revision level.

Thus, in the following complete part number,

093-000114-00

the initial field indicates an 093 category of manual, "software user or reference manual". The specific part number identifies the *RDOS/INFOS System User's Manual*. This manual is at revision 00.

Data General currently defines six general categories of manuals:

- 012 Catalogs, price schedules and other sales information.
- 014 Hardware technical reference. Manuals in this category contain initial, condensed hardware information. Hardware technical reference manuals are often, but not always, replaced by more expansive manuals in the 015 category. No manuals in the 014 series are described in this bibliography since their topics do not include software documentation.

- 015 Hardware technical manual. Manuals in this category describe at length the hardware aspects of Data General computer systems. Included in this category are manuals describing the machine language instruction set of ECLIPSE computers, and a manual used in peripherals programming.
- 085 Release Notices (described in the previous section).
- 086 Addenda (also described in the previous section).
- 093 Software reference or user manual. This is the largest category of software documentation.

Keeping in Touch

If you have any comments or criticisms you'd like to make about our documentation, please let us know. You may write to us directly by using the Remarks Forms found in the backs of our manuals, or you may contact your local sales or service office. Please use the Remarks Form only to comment on manuals. To report software difficulties or order additional products, contact your local Data General representative.

End of Manual



Data General Corporation, Westboro, Massachusetts 01581, (617) 366-8911