VIDEOTEX AND OFFICE INFORMATION SYSTEM

Version 1.1

CUSTOMER GUIDE

March 15, 1984

Contents

3/Maintaining The VIS Database

What Is VIS ?

The **Videotex and office Information System** (VIS) is an expertly designed information storage, retrieval and distribution service that provides a variety of information to remote users via standard telecommunications methods.

VIS distinguishes itself from existing Videotex systems in a number of ways, not the least of which are its cost effectiveness, its adaptability to virtually any information distribution environment, its flexibility in terms of application, hardware components, and configuration, and its versatility and ease of use.

VIS is Cost Effective

Traditional computer applications usually involve costly technical personnel and machine resources throughout their development, programming, and maintenance, and frequently yield limited and typically short-lived results. VIS cuts these costs down dramatically by providing a flexible, self-contained database management mechanism that keeps the technical aspects of data storage and retrieval transparent, while the practical aspects of data entry and maintenance involve a simple yet versatile set of easy-to-use tools and procedures.

The entire process of managing a VIS database can be handled by an individual who first understands the information needs his service addresses, and then, optionally with the aid of data editing or word processing personnel, directs VIS in the performance of the tasks that meet these needs. The advantages in terms of labor costs and quality of service should be obvious.

VIS is Versatile

The versatility of VIS lies in its inherent applicability to almost any definable information service environment. The two basic types of information services, e.g. PIVS and MMS, though they address distinctly different needs, expand their capabilities and usefulness into new dimensions when brought within a VIS framework.

The **Private Internal Videotex System** (PIVS) services the information storage and delivery needs of a single business entity. A firm's computer department or information services group can easily provide many company employees with internal ("in-house") information, such as financial data, sales and inventory figures, personnel records, etc.

The **Mass Market System** (MMS) seeks to serve a wider range of information needs. The VIS "public" database stores a variety of information items, and access is provided to a diverse user base on a subscription basis. In MMS modes VIS carefully secures a number of distinct applications simultaneously.

The Contents of VIS

Typically, most of the information in a VIS database is of a textual nature, like news, sports and weather reports, price lists, stock summaries, and advertising copy. VIS, however, provides the capability to store and "download" programs, such as games, for execution at the remote site. Add also the facility to store and transmit colorful pictures, complex diagrams, plans, and other graphic images to a user terminal and the potential uses of VIS become limited only by one's imagination!

A complete list of the possible contents of a VIS database is **IMPOSSIBLE!** Information needs and services are expanding and evolving at a staggering rate. It is possible, however, to identify categories into which many services can be grouped, and name just a few of the instances where VIS can produce practical and profitable results.

Archival ("static") Information Retrieval Services - including legal information used in case searches; medical information, such as patient records or diagnostic aids; pharmacological databases containing information on drugs, medicines or vitamins; insurance industry data for policy or client searches; mortgage and title search facilities based on sales records; library card catalogs; biblical or encyclopedia research databases; record, tape and book lists and catalogs; name and address, phone directories and searches; inventory and catalog lists; information and listings for special associations (FFA, AQHA, livestock, farming, bowling, etc); publication and abstract indices covering general or special areas of interest.

Continuously Updated ("dynamic") Information Services - including agricultural data services; periodicals, newspapers, magazines (news, weather, sports, etc); Chamber of Commerce data and city information and locational data; Better Business Bureau listings and information; real estate data for multiple listing services; mailing lists; credit bureau information; Stock Market quotations for Commodity Exchanges, bonds, OTC, etc; classified advertising; "Yellow Pages" with current promotional data; "help wanted" or "available for work" listings.

Specialized Information Services - including computer program and game ("arcade") systems; "graphics" systems for producing pictures, maps, graphs and charts; electronic distribution of computer program libraries; Government databases such as auto licenses and titles, tax rolls, court case records, voter rolls and criminal records; educational records used for enrollments, class schedules and student information.

VIS is Dynamic

You can count on VIS evolving right along with the needs of an information-conscious world. Don't be surprised to see some of the following VIS applications in the near future:

Office Automation - may include electronic mail services; bulletin boards; electronic "filing cabinets" for letters and memos; office calendars and schedules; filling out forms electronically.

Interactive and Computational Information Systems - may include catalog sales encompassing item searches, detailed specs, and orders; estimating services used in engineering, architecture, and construction; project time planning and management; consumer surveys, market planning and research; reservation systems for airlines, hotels, cars, and entertainment events.

Specialized Information Services - may include electronic auctions of livestock, equipment, and estates; "gateway" services to other information systems such as Prestel, Telidon, CompuServe, Dow-Jones, etc; "citizen's band" crosstalk services; electronic banking services; financial analysis services; interactive student counseling databases.

VIS is Easy to Use

The data access method of VIS is one of its most impressive features. Most computer applications tend to dictate the terms of user-machine interaction, using explicit, jargonistic vocabularies, and requiring strict, almost dogmatic syntax. VIS is quite the opposite: requirements are few, user conversation is friendly and natural, and request processing is accommodating yet efficient.

The procedures and tools used in VIS database maintenance are straightforward, easily mastered, and easy to use. This guide, the VIS CUSTOMER GUIDE, is devoted to the area of VIS database management and the tools and techniques to make the job efficient and effective.

The VIS Participants

Operationally, VIS participants can be clearly divided into three functional groups. Each makes use of its own facilities and carries out its distinct activities and responsibilities. Within either a PIVS or MMS environment, the VIS participants are the same. For the sake of convenient terminology, we refer to the three operaticnal entities in the VIS network as the System Operator, the Customer, and the User.

System Operator

The **System Operator** is the information service manager. He installs and maintains the VIS hardware and software. From the initial installing, configuring and activating of the VIS equipment and programs to the periodic system "tune-ups", backups, "reorgs" and upgrades, the System Operator ensures the continued and reliable operation of the VIS network.

If the System Operator directs his service to a mass market rather than a private user base, he also arranges to provide VIS facilities to Customers, and he is responsible for defining and maintaining all the information about them.

Customer

The **Customer** is the information provider. He uses the facilities of the VIS Host to store and distribute items of information, or Documents, to remote Users. He is responsible for the content and presentation of his Documents.

In a PIVS environment, the Customer can be the System Operator himself, or he can be an "in-house" data editor or word processing director. In an MMS environment, a Customer can be one of several independent entities sharing the VIS resources.

The VIS Customer decides to whom and on what basis his Documents are to be made available, and he enters and maintains the data about his Users, including names and addresses, logons, security levels, and terminal types. Periodic problem tracking, utilization analysis, and in some cases billing are among the Customer's other activities.

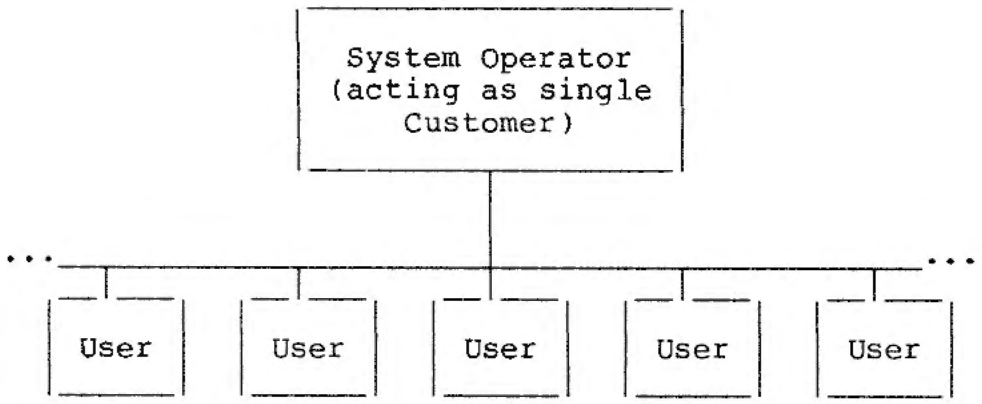
User

The **User** is the information consumer. He (or she) "dials up" the VIS Host and "logs on" to VIS from his remote site. During his session with VIS, the User requests information by entering the names or topical references of items stored in the VIS database. He can request single items or he can access groups of related Documents as selectively as his requirements dictate.

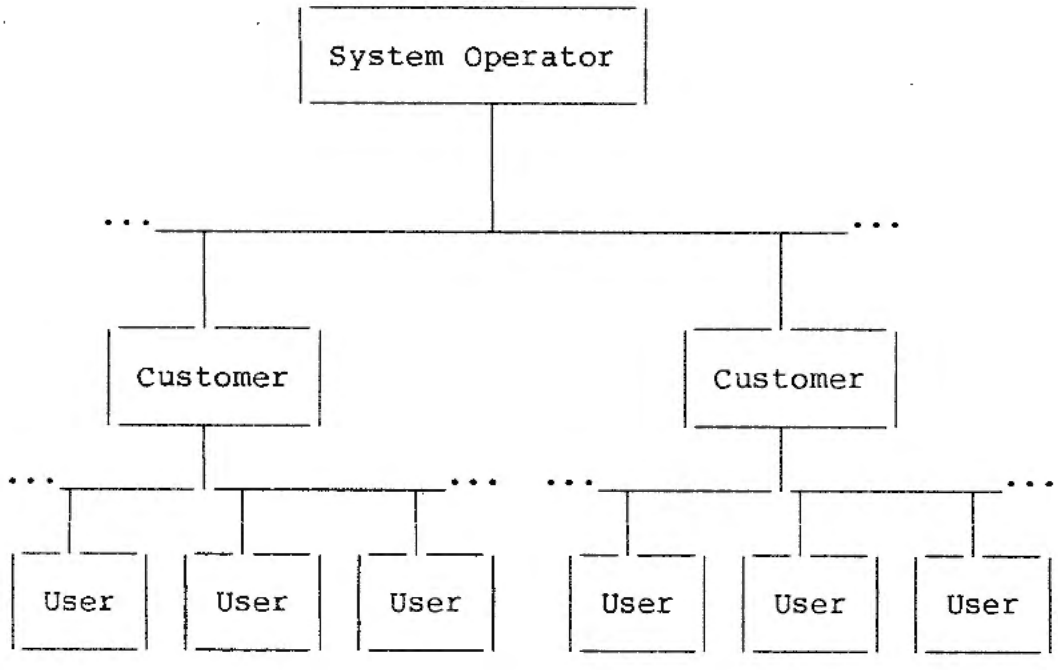
In general, a User's requests yield textual material, but if his terminal device and its controlling software support it, he can also receive graphic images during his request sessions. In some cases, programs can be "downloaded" to the User terminal for immediate execution or local storage.

The User ends his session by "logging off" and/or hanging up the phone line.

The following diagrams show the "typical" relationships between VIS participants:



Typical PIVS Relationship



Typical MMS Relationship

The VIS Network

From the bottom up VIS has incorporated some of the latest Tandy hardware components into a number of efficient yet highly flexible configurations. The three VIS groups function as distinct entities "physically" as well as operationally. Each makes use of its own hardware components to accomplish the task at hand.

Again for the sake of convenient terminology we shall refer to the three parts of the VIS network as the Host site, the Customer site, and the User site. These facilities can be at a single location, or in entirely separate places, or a combination of both.

Host Site

The **Host site** consists of the TRS-80 Model 16 microcomputer with one or more "hard" disk devices on-line. These two components are the heart of the VIS network.

The Model 16's advanced features and capabilities, and the large capacity and rapid access facilities of the "hard" disk unit(s) are managed by the TRS-XENIX operating system, a powerful multi-tasking, multi-user operating system derived from UNIX.

Customers and Users tie into the Model 16 through one or both of its standard I/O "ports". In "basic" VIS configurations both ports handle User calls through 300 baud modems. In more "advanced" systems the TRS-80 Communications Multiplexer (a fully self-contained, processor-driven "electronic funnel") can be attached to one port to handle up to 16 separate User calls at the same time, while Customers interface through the other either directly or through a high speed modem. A "super" system configuration includes a TRS-80 DT-1 Data Terminal for multiplexer status and diagnostic inquiries.

Customer Site

The **Customer site** can be a single "local" or "remote" work station, or, as in the case of an extensive MMS service, one of several fully staffed locations. Data preparation is usually carried out on a TRS-80 Model 12 microcomputer.

A self-contained, full-featured microcomputer in its own right, the Model 12 can be tied to the Host either directly as an "on-site" work station, or it can operate as a "remote" data preparation and collection point. In this latter case, periodic "uploading" of VIS materials to the Host processor involves a high-speed modem and phone line.

The "software" used for preparing VIS materials can be either a word processor (typically Scripsit), or, in some cases, the Host operating system's text editor. Other devices and software may be supported for the purpose of data editing and Host interaction; contact the VIS information service manager for details.

User Site

Flexibility at the **User site** is a key asset of VIS. Users can "dial up" to the VIS Host with any of a wide variety of terminal devices. Although certain devices are recommended, any device that can transmit and receive ASCII 7-bit characters over standard RS-232 asynchronous communication lines can be adapted for use.

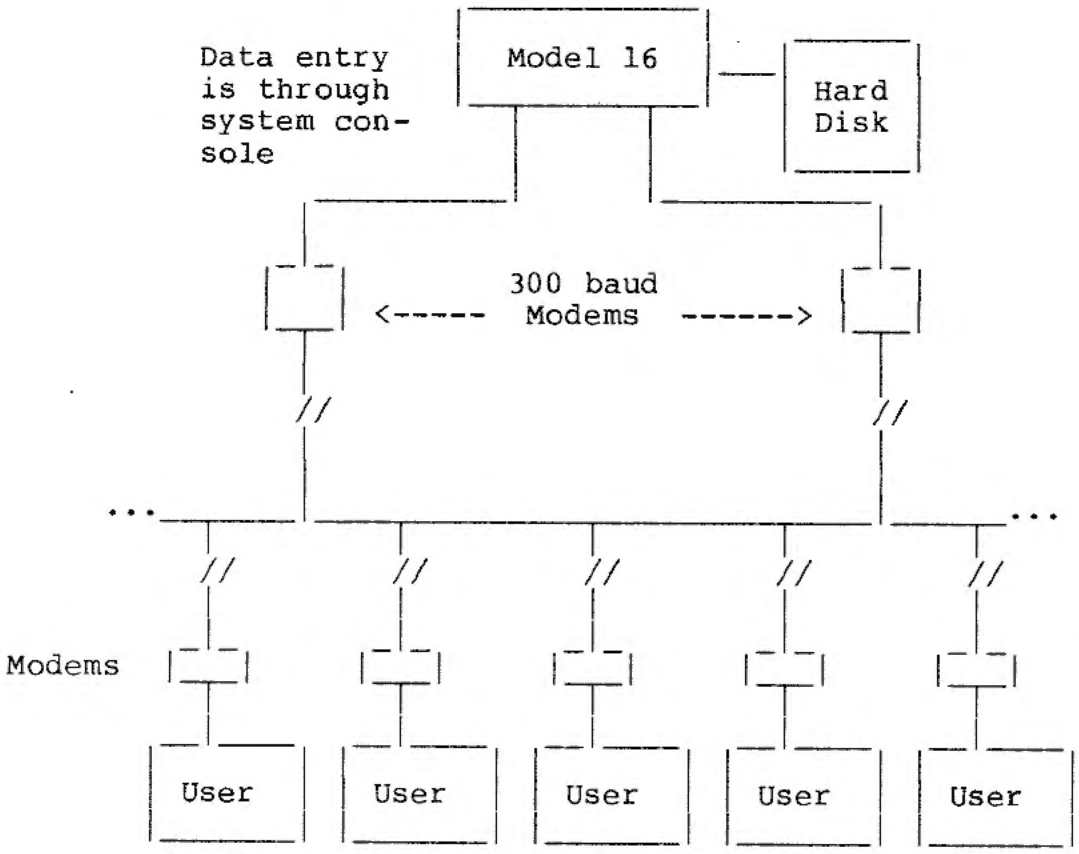
There are several characteristics and capabilities that a terminal device can possess in addition to the ability to transmit and receive characters. You should be aware of the "typical" devices your Users employ and the characteristics these devices exhibit. Each terminal's capabilities determine how and to what extent VIS interacts with it.

VIS, being a Tandy product, is designed to be most compatible with other Tandy equipment. The recommended User terminal devices are listed below.

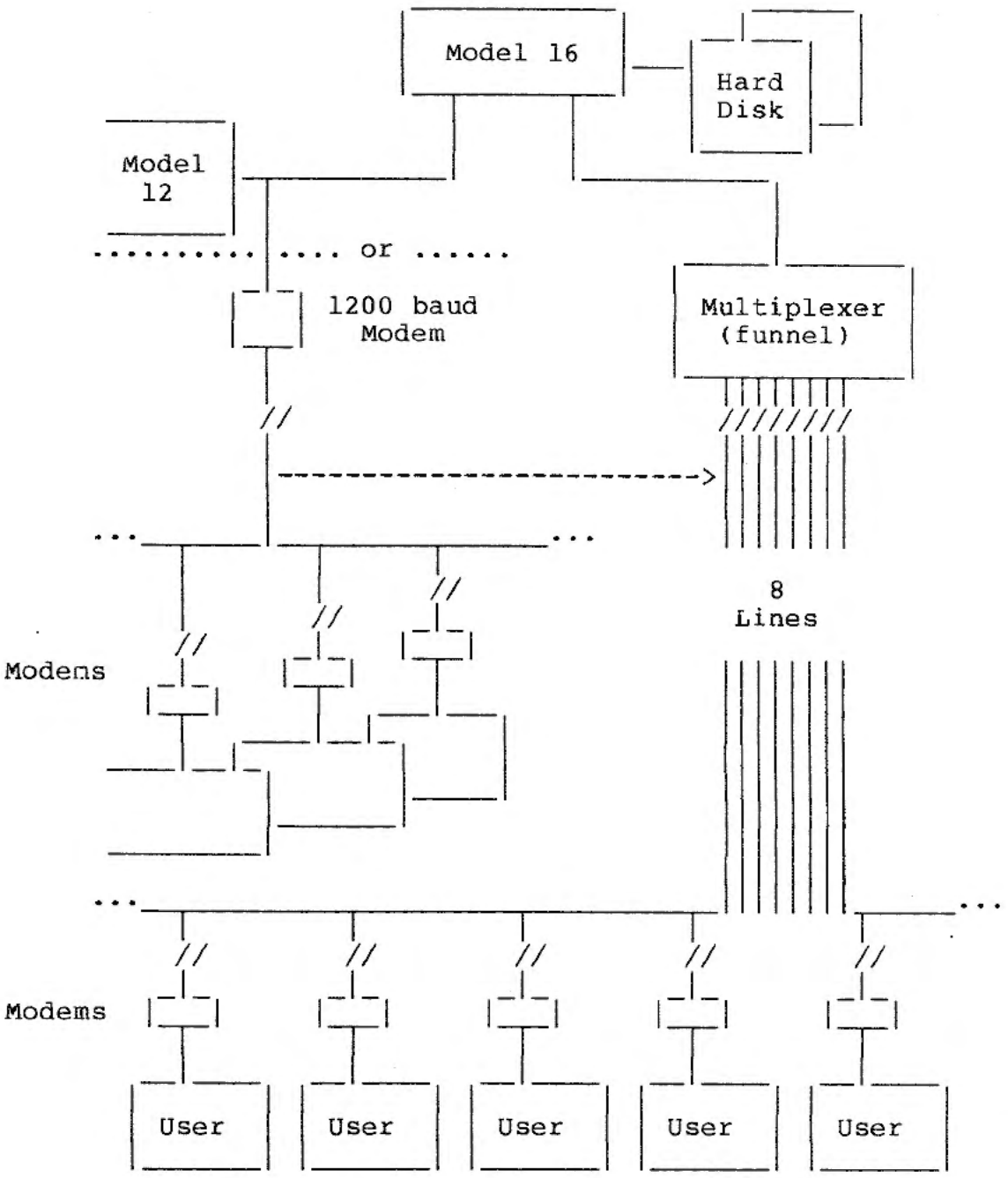
TRS-80 DT-1 Data Terminal
TRS-80 Videotex Terminal
TRS-80 Color Computer
TRS-80 Model I
TRS-80 Model II
TRS-80 Model III
TRS-80 Model 4
TRS-80 Model 12
TRS-80 Model 16
TRS-80 Model 100
TRS-80 PT-210 Portable Data Terminal
TRS-80 Tandy 2000

All of the devices shown here (except the DT-1 and PT-210) are microcomputers and must run either Videotex or terminal "emulation" software when used to interface with VIS. All but the Videotex Terminal, PT-210 Paper Terminal and Model 100 computer require external modems to communicate with the VIS Host from a remote location.

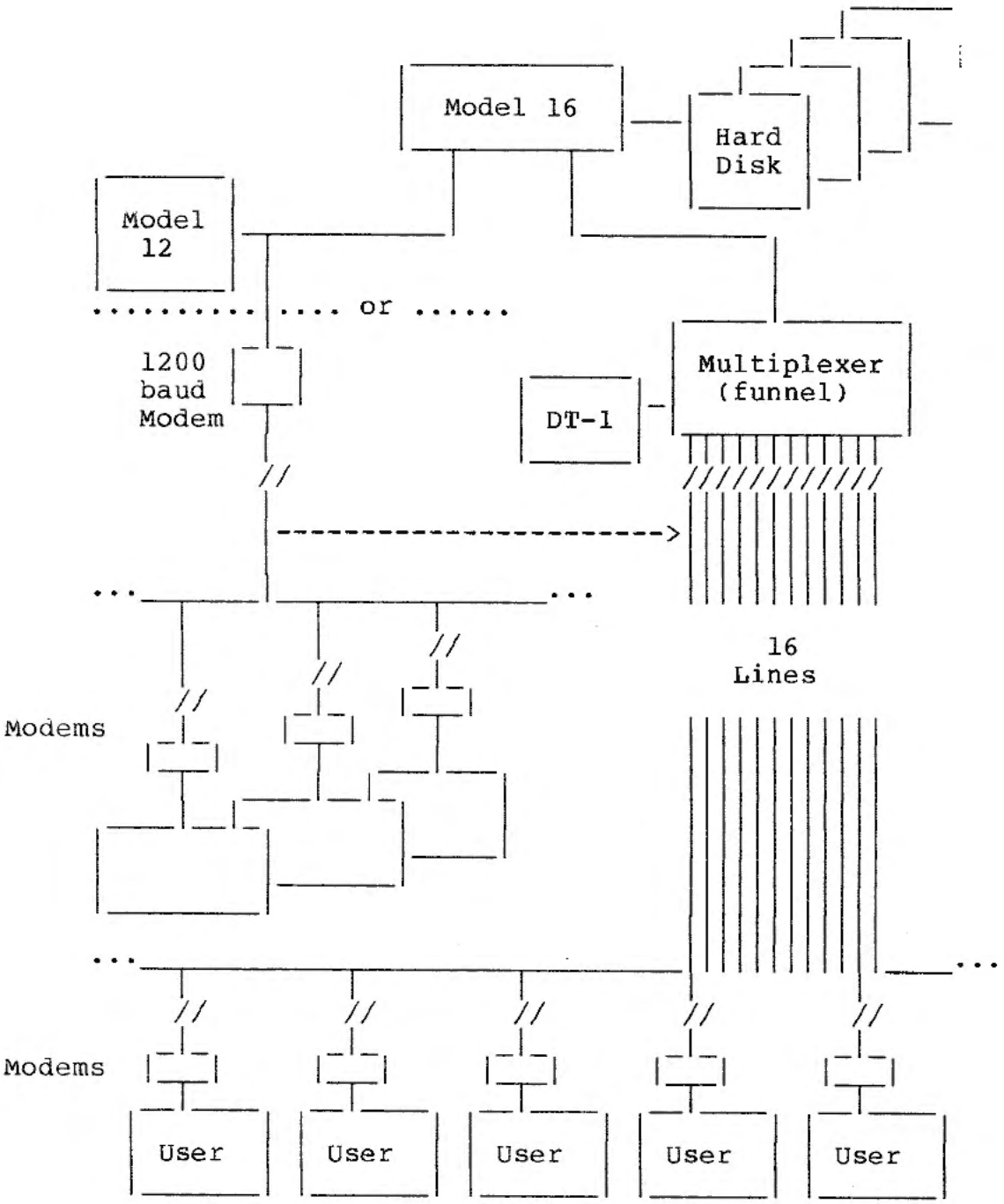
The following three diagrams show how the VIS network might be configured:



A "Basic" System



An "Advanced" System



A "Super" System

Connecting to VIS

There are three ways for a user to become connected with VIS: **direct connect, modem connect,** and through a line on a **TRS-80 Communications Multiplexor.** Which of these options are available is dependent on what the system operator has implemented and which VIS system has been obtained (Basic, Advanced, or Super). What the user needs to know, is the proper baud rate to communicate with VIS.

With a direct connection to the VIS console, a user may communicate at several different baud rates: **300, 1200, 2400,** and **9600** baud. Users must be informed by the system operator as to which rate is appropriate so they can set their terminals accordingly.

From a modem connect, only two baud rates are available: **300** and **1200** baud. Moreover, once a user has established a connection over the phones, it is necessary that they strike the <ENTER> key a series of times to complete the communication link. This allows VIS to automatically determine whether the remote site is communicating at 300 or 1200 baud.

Communication is simplified when going through the multiplexor. As **300** baud is the only baud rate available, all that is required is to set the remote terminal also to 300 baud and make the call to establish the connection with VIS.

The Customer Processing Environment

The VIS Customer is responsible for populating and maintaining the VIS database. He provides the information (Documents) that VIS stores and distributes to his Users. The facilities provided for this task are easy to use and yet powerful. The entire process lends itself well to routine and yet provides flexibility and latitude for creative expression.

Maintenance usually refers to the work of keeping a mechanism in good operating condition. In the case of the VIS database, maintenance takes the form of adding, changing, and deleting information entries (Documents). It is the job of the VIS Customer to maintain the VIS database.

All the activities involved in maintaining the VIS database can be summarized in the following sequence of events:

1. Materials are prepared for processing.

2. These materials are supplied to VIS.

3. The materials are processed and applied to the database.

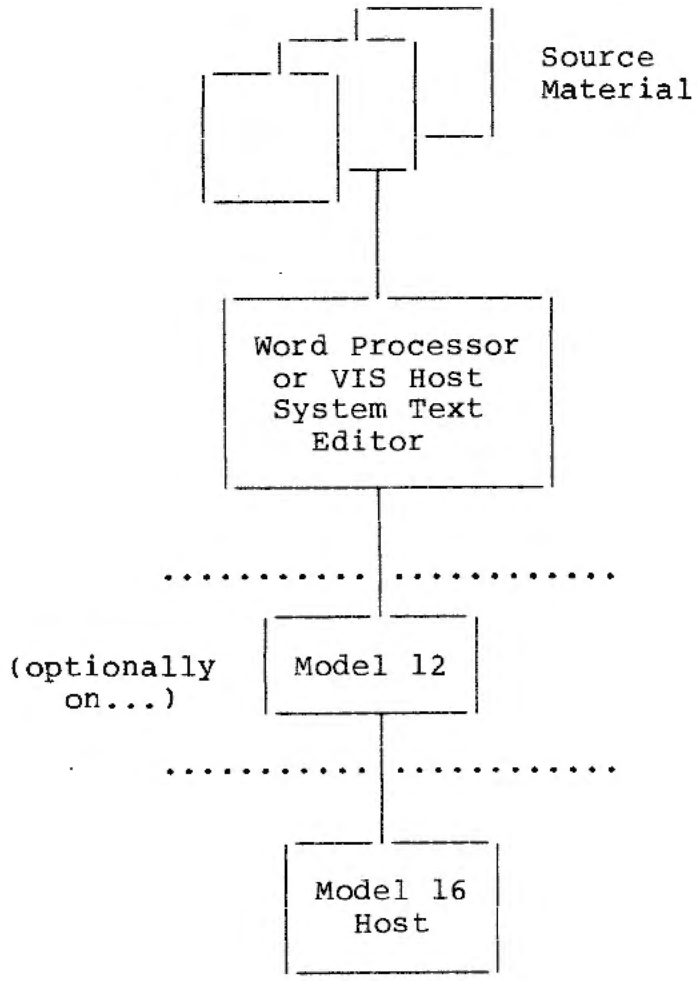
4. Results are examined and/or printed, and corrections are made.

Each of these steps is discussed in detail shortly.

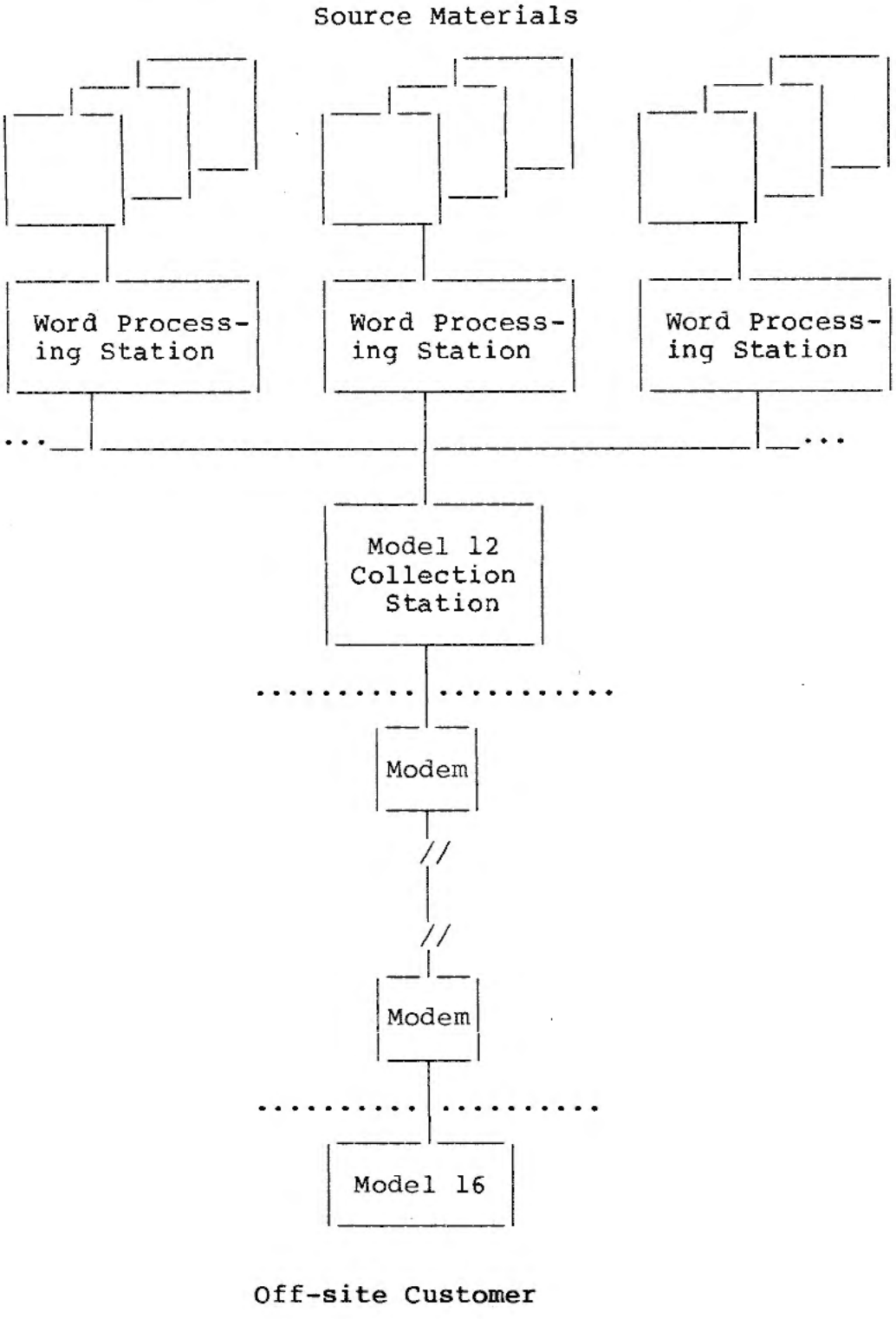
As mentioned earlier, the job of VIS Customer can be carried on either at the Host site, or at a remote location. In the former case, the role of Customer can be executed by the System Operator himself using the system console as his work station, or by one or more additional personnel using a TRS-80 Model 12 attached to the Host Model 16 as a separate work station.

In the latter situation, Customer operations can take place on a grand scale. An entire staff of word processing and copy editing personnel can work at several stations, and at regular intervals the accumulated materials can be transmitted to the VIS Host for processing.

The following diagrams illustrate the range of possible Customer site arrangements:



On-site Customer


Source Materials
Word Processing Station
Word Processing Station
Word Processing Station
Model 12 Collection Station
Modem
Modem
Model 16
Off-site Customer

The VIS Database

The VIS database consists of a variety of data items known as **Documents.** Each type of Document is designed to accommodate the storage and, in most cases, transmission of a particular type of information.

Certain Documents store the "master" information that identifies the various VIS participants. Others hold data that describes the way items within the database are logically related. The rest store the actual text and other material that is transmitted to your Users during their VIS sessions.

A User session consists of a series of "requests" for information entered on the terminal and the viewing of the results on his terminal screen. The session ends when the User "logs off" and disconnects the phone line. The quality of this session depends on the quality of your Documents and the care and creativity you, or your personnel, exercise in preparing and referencing them.

Following are several discussions about the various VIS Documents.

User Master and User Detail Documents

The User Master Document contains the name, address and telephone number of a specific User. You must define a User Master entry for each of your Users, making sure that their ID designations are unique (ID= specification in Verb).

User Detail Documents contain the information that defines your Users to VIS. Included in this entry are logon ID, password, terminal type, security status, YTD and PTD dollars spent, etc. You must create at least one User Detail Document for each User, although a single User can be defined by several. Each User Detail entry defines and sets a different mode or level of access with respect to VIS. At logon time, the ID and password specified by the User identify the particular User Detail values that are to apply during the session.

When a User "dials up" to VIS, he is greeted with a system logon screen and prompted for a logon ID and password. These values are the ones that VIS uses to identify a User and the Customer to whom he belongs. You, as a VIS Customer, are responsible for establishing the unique ID and password values for your Users. A successful logon begins your User's session with VIS.

Text Documents

Text refers to normal written material. This manual is an example of text; so are letters, lists, news and weather reports, menus, articles, etc. Text Documents can consist of multiple pages of information displayed one screen at a time. There is no restriction, other than that imposed by the availability of on-line storage, as to the maximum size of a Text Document; the smallest Document, however, occupies no less than 128 characters worth of storage.

The contents of your Text Documents is the heart of your information service! Their subjects, their interconnection, and their format (aesthetics in large part) are entirely up to you and are guided by the information needs your service addresses, the sophistication and interests of your Users, and your own judgement and creativity.

In full recognition of the fact that the presentation of Documents is no less important a consideration than the material they contain, VIS provides a number of functional tools that direct the formatting, sequencing, and interactive processing capabilities of your Text Documents. These tools are in the form of special character-code specifications that are included as part of the Text itself. VIS interprets these sequences at transmission time and processes them accordingly. This feature of VIS is known as Text Programmed Presentation and is discussed in detail as a major topic of its own.

Program Documents

A program is any set of instructions designed to perform a specific task or series of tasks. Though programs frequently resemble text, they must be interpreted or "compiled" by a language translator before a computer can execute their instructions. VIS stores programs as Program Documents and "downloads" them to compatible remote terminal devices. VIS downloads these programs using CompuServe's B Protocol.

Some terminal devices, typically microcomputers under the control of Videotex software, are capable of receiving and executing the contents of Programs Documents within their own environments. Control codes sent as part of the "downloaded" data cause it to be loaded directly into memory and executed immediately. Programs "downloaded" in this fashion are called "arcade" programs.

Some devices can store "downloaded" data on their local storage media (usually diskette) for future use. This "disk transfer" process is also guided by special control character sequences included as part of the Program Document's transmitted data.

The content of a Program Document can be the "source" statements of a program to be compiled and executed at a remote site ("disk transfer" mode) or the "object" code produced by the assembly or compilation of source statements or instructions that are to be executed immediately upon reception at the remote site ("arcade" mode). These contents must be stored as separate files.

How you obtain the Programs you store and distribute to Users is not discussed here. What your stored Programs do or what they provide is determined by the nature of your particular service. There are a few considerations, however, that apply to Program Documents and their use within the context of VIS.

Once a "source" or "object" file has been created, it must be prepared and stored in the form in which it is eventually to be transmitted. "Arcade" Programs must contain the correct control information to direct the loading and execution of the program. "Disk transfer" Programs must include the information necessary to direct the storing of the program onto the local diskette device. The means to accomplish this formatting are discussed in the section on "The vis.preedit Utility".

Not all User terminal devices are capable of receiving and executing Programs. As mentioned earlier in the discussion of Terminal Types, only certain terminal devices, i.e. microcomputers operating as terminals, have the resources to accept "downloaded" source and/or object code and execute it within their own environments.

In addition, an "object" Program assembled or compiled for execution on one type of microcomputer can rarely be processed by another. Each microcomputer is a configuration of input and output facilities directed and managed by a particular type of "microprocessor", and each microprocessor exhibits its own capabilities, characteristics, and limitations. The TRS-80 Models I, II, III, and 12 employ the Z80 microprocessor; the TRS-80 Color Computer and Videotex Terminal use the 6809E "chip". Object code that runs on one will not run on the other.

Each Program imposes certain requirements in terms of memory and I/O resources needed for its execution. These additional restraints must also be considered when making Program Documents available to Users.

You can restrict the transmission of your Program Documents to compatible devices by specifying the requisite characteristics in the ONLY= specification of the Document's defining Verb. Only terminals meeting these requirements, i.e. their set of characteristics determined at logon time includes those specified, will receive the Document.

Another practice is to define a "descriptive" Text Document outlining the characteristics, special requirements, and send/receive procedures that apply to the Program requested by a User. The User receives this Document first, and if his environment is compatible, he can, under Menu control (to be described later in connection with Text Documents), either continue with, or bypass altogether, the actual transmit/receive process.

Graphic Documents

The term "graphic" usually refers to a screen image of something other than one of the letters, numbers, or special characters that make up the standard ASCII character set. Color reproduction is included in this distinction as well. Designs, colorful pictures, charts, graphs, maps, and various geometric figures are examples of graphic images.

Many terminals are capable of producing "high resolution" images, i.e. images that reference and manipulate the individual "picture elements" of the terminal screen. Some produce only "medium" resolution graphics - images based on small blocks of picture elements. Others offer a special set of graphics characters, or "coarse graphics", with which to produce "low" or semi-graphic images. And some terminals have no graphics capability at all, but can be color controlled.

You can store the character sequences that produce graphics as VIS database entries and VIS sends them to compatible terminal devices.

Again, with respect to terminal devices, not all are capable of generating graphic images. As with Program Documents, transmission of Graphics can be restricted by using the ONLY= specification in the defining Verb. Terminals that do not include your specified code(s) as part of their definitions will not receive the Document.

The particular codes and sequences that produce graphic and color images vary from terminal to terminal. You will have to consult a specific terminal's documentation on this subject to find out how to manipulate its graphics capabilities to suit your purposes. The actual graphic "strings" can then be stored as Graphic Documents.

There is an important item about designing graphic documents for VIS that should be mentioned here. Though codes differ, almost all terminals will have a code that changes the terminal from text mode to graphics mode and another code to return to text mode. It is necessary when designing a graphic document that the last code in the document be that which changes the terminal back to text mode. If this is neglected, it is possible that a User who requests a graphic document could be left in graphics mode indefinitely until VIS disconnects his communication line due to an excessive amount of time with no input from the terminal, (VIS ignores terminal input in graphics mode).

Input Documents

The Input Document provides a place to store a VIS request line. When an Input Document is accessed or invoked, VIS processing continues just as though the characters of the Document had been typed by the User and submitted to VIS as a normal request.

The "language", special characters, access facilities, etc involved in making requests are covered in detail in the VIS USER GUIDE.

Initial Input Documents

When a new User is allowed access to the data base, they should have an "initial" Input document that is associated uniquely with them added to the data base at the same time. This initial document is used to store a User's "default request", (for more information on the "default request", see the User Guide), and it is requisite upon a Customer to add an initial document for each of their Users. If there is no initial document, it is impossible for a User to use the default command ('+') to set a default request.

The initial document is constructed just like a normal Input document, (see the sections on "Maintaining the VIS Data Base"), with the exception of the NAME. The NAME of the document must consist of a concatenation of the User's Logon ID and his password, followed by the word "INITIAL", (e.g., "P1111ABCDINITIAL"). For this reason, each User will have his own, unique, initial Input document. For an example of an initial document, please see the section on "Sample Verbs".

The Text part of the initial document contains the actual default request. When setting up an initial document for a User, a Customer can include any default request desired. However, since it is unlikely that a Customer will know all of a User's preferences, one suggestion would be to leave this section empty (see the next section for a description of a "null" document).

Null Documents

It is possible to insert a "null" document into the data base, i.e., a document place holder with no document associated with it. There are a number of ways this feature can be useful.

First, in conjunction with the previous section on "Initial Documents", null documents can serve as initial documents. When a User hits <ENTER> without typing something after the "VIS:" prompt, he will merely receive a new "VIS:" prompt instead of a superfluous document. Similarly, a User may wish to re-set his default document to a null document at some point, to zero out his default document without setting it to something new.

And, from a Customer's point of view, it may be necessary to reserve a document name for future use. Giving a null document that name could serve this purpose. Also, if a Customer has constructed a Menu document that refers to various other documents, but has not yet had time to generate all of the documents, the null document can serve as a placeholder for the missing ones.

Logon Document

The Logon Document is the first document a User sees after he logs on to VIS. This text document should be designed by the VIS Customer to fit his needs. It may be informational in nature, an advertisement or just a "null" document.

The Logon document is constructed just like any normal text document with the exception of the NAME. The NAME of the document must consist of the Customer"s five character **(/ocode)** and the word "LOGON", (e.g., "PLANTLOGON"). This unique NAME is the means for VIS to display this document when the User first logs on to VIS. For an example of a Logon Document, please see the section on "Sample Verbs".

Logoff Document

The Logoff Document is the last document a User sees when he logs off of VIS. This text document is required and may be designed by the VIS Customer to fit his needs. It may be informational in nature, an advertisement or just a "null" document.

The Logoff document is constructed just like any normal text document with the exception of the NAME. The NAME of the document must consist of the Customer"s five character **(/ocode)** and the word "LOGOFF", (e.g., "PLANTLOGOFF"). This unique NAME is the means for VIS to display this document when the User logs off of VIS. For an example of a Logoff Document, please see the section on "Sample Verbs".

Document Security

Depending on the sensitivity of the material contained in certain Documents, you may want to restrict their availability to particular Users. The VIS security feature provides this capability.

Security Codes

You assign each User a Security Code. You also assign one to each Document. The code consists of two components: one designates the Security Group, the other sets the Security Level within the group. Security Codes are set when Users and Documents are entered into the system. The codes can also be changed through normal maintenance procedures.

Security Group

A **Security Group** is a value from 0 to 15 that identifies the general category of security to be associated with a User and certain Documents. In the case of groups 0 through 14, a User's Security Group must match that of the Document he wishes to access. A group 15 security, however, permits access to Documents within any of the other groups.

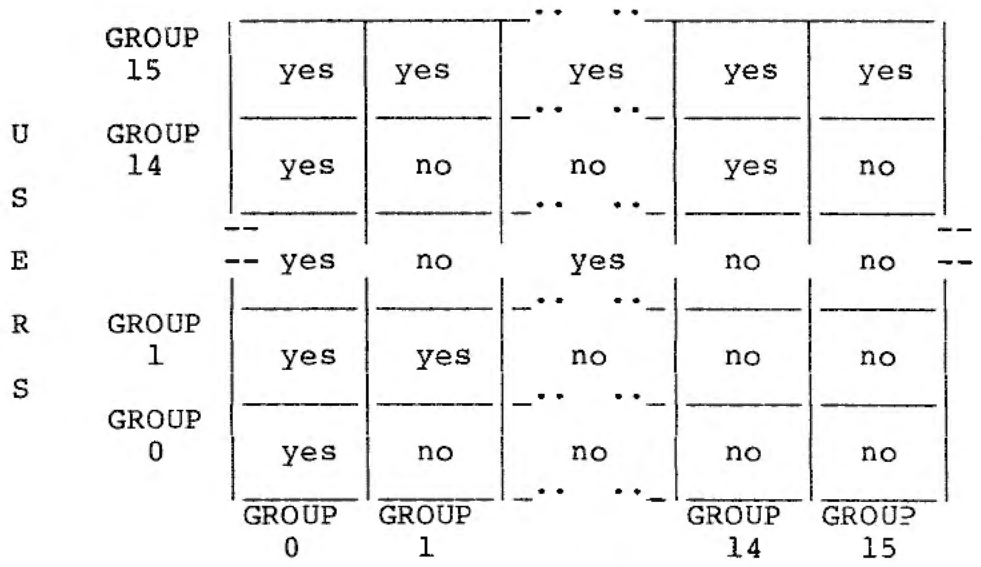
In the case of Documents, however, groups 1 through 15 will be group-restricted. Group 0 Documents will be available to all groups.

Security Level

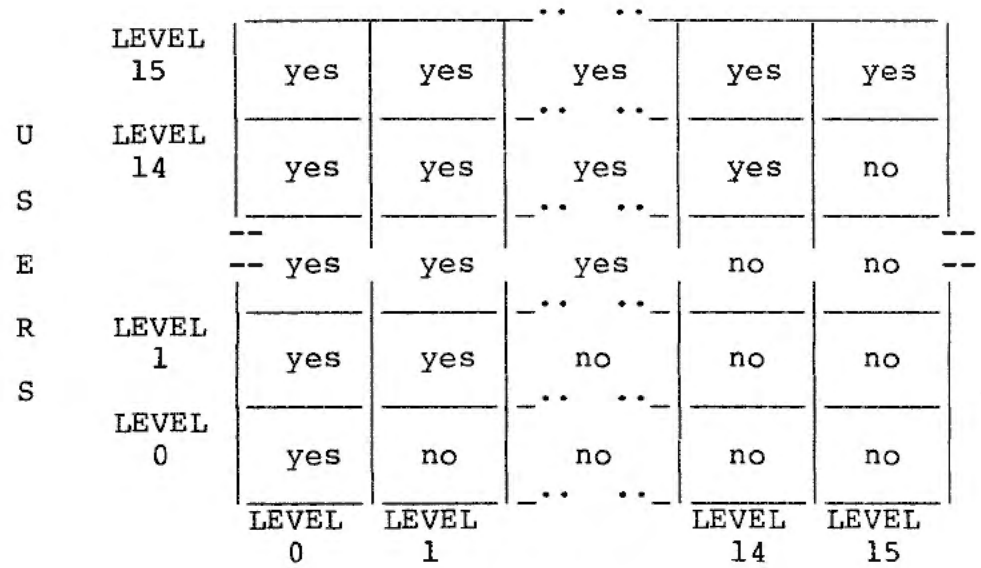
In addition to the 16 group designations, there are 16 values, 0 through 15, that define the **Security Level** of Users and Documents. A User with a Security Level that does not equal or exceed that of the requested Document is denied access to that Document regardless of his or the Document's Security Group designation.

A total of 256 security designations are available, offering a wide range of possible security configurations. The following charts show the Security Group and Security Level relationships that can be implemented using the VIS Security Code facility.

Note that Security Groups are matched first. Where specifications are compatible there is a "yes"; where they are not, "no". Security Levels are checked next. Where the User's level matches or exceeds that of the requested Document, access is permitted as indicated by "yes"; otherwise access is denied as indicated by "no":



Security Group Relationships



Security Level Relationships

Timeouts and Retries

Timeouts and retries, in general, are to be regarded as the only way that VIS has to ensure maximum availability of ports to its users. These mechanisms are designed to prevent the excessively slow, unattentive, or malicious User from monopolizing a port's resources to the exclusion of other Users. Though the timeout and retry facilities differ in function, they both may result in "disconnection" of a User line. In the case of a "direct connect" environment, the process of physically disconnecting (hanging up) a phone line is impossible. Instead, the system merely starts over again at the "ENTER LOGON ID:" prompt. If at any time the User disconnects the line, VIS will log the User off and free the line for another user.

Timeout values are expressed in tenths of seconds and specify the interval of time during which VIS waits for a User response. Note that VIS marks the interval from the moment the last message was sent. A message consists of a string of characters, null or otherwise, terminated by an <ENTER> character. All other characters received during the timeout interval will not reset the timeout mechanism, except of course the <BREAK> character.

Retries specify how many times a User will be re-prompted for a certain input request before the system assumes that no correct response will be forthcoming and takes the system default action, i.e., logging off and hanging up on the User.

Timeouts are of two basic types. The first is the "no valid logon" timeout. This timeout assumes that a caller has not correctly received the "ENTER LOGON ID:" prompt or has responded to it but the response was not received by VIS. Until a User has logged into VIS, it is unknown which Customer he might desire. Thus, during the logon process the timeout and retry values are taken from the System Operator's Customer record. As soon as the User has completed logon, the User Detail values for timeout and retries are used. When the "no valid logon" timeout occurs, the "ENTER LOGON ID:" prompt is sent again. This process will continue until the number of retries specified in the System Operator's LOGON-RETRIES parameter is exceeded. If a valid logon has not occurred within this number of retries, an error message is sent to the user and the line is disconnected and freed for another User. If a valid logon does occur, then this timeout/retries is no longer in effect.

The second timeout is the "user went to sleep" timeout. This timeout is defined as a specified time limit following the "VIS:", "Which?", and "Load?" prompts for input. This timeout is set by a Customer via the TIME-OUT-SECONDS parameter of the User Detail record. If this timeout occurs it is retried up to RETRIES (again from the User Detail record) times before an error message is issued. If it was a "Which?" or "Load?" prompt which was timed out, then the User is sent a "VIS:" prompt and VIS waits for input. If it was the "VIS:" prompt which timed out, then the User is logged off and the line is disconnected.

Both timeouts and retries are governed by variables set by the Customer for each of his Users. It is important to note that these values are associated with the User, not with any specific port. Note also that the Customer can set a User's timeout value to 0, which will tell VIS NEVER to timeout that particular User. In this way, the User can be identified as "privileged" and will not be forcibly logged off the system as a result of timing out.

Values for timeouts are stored in the Customer and User Detail records. The Customer record provides default values at the time a User Detail is added, i.e., if the Customer does not supply a value for the TIME-OUT-SECONDS parameter at the time a User Detail record is ADDed, the TIME-OUT-SECONDS value from the Customer record will be used as the value. In the same way, the System Operator record supplies the default values during Customer record creation. This is to allow the System Operator and Customer to pre-set their respective Users for certain benefits/disadvantages in keeping with their mutual contract.

Topical References

Document retrieval begins with a User request, which is composed of one or more words followed by a carriage return (**<ENTER>**). The request can be for a specific Document or a group of topically related Documents. The words that gain access to Documents are collectively referred to as Topical References, and are established and maintained by you as a VIS Customer. Topical References allow a User's requests to be very casual and yet meaningful.

The interrelationship of the references that apply to and connect your Documents (the **reference structure**) is entirely in your control. The extent to which you "cross reference" your database items, and the care you take in choosing the words that comprise this reference structure will have a profound effect on the ease of use, convenience, and "friendliness" that your service presents to your Users.

You should try to devise a Document reference structure that exhibits not only logical clarity and accuracy, but also sensitivity to the spontaneous approach that most Users will take in selecting the words they use as requests. Your reference structure should always be undergoing change, refinement, and hopefully improvement.

Titles

Every Document has a name, or **Title**, that uniquely identifies it. Each Document can be accessed directly by its Title.

Titles consist of up to 26 characters, including letters (A through Z), numbers (0 through 9), and, for the sake of readability, occasional periods ("dots"). (DO NOT define a Document with the Title "..." (three dots)! An attempt to access this entry will log the User off VIS.) Any other special characters are not allowed in Titles.

Consider a VIS database dedicated to articles (Documents) on the local vegetation of a particular geographic area. A partial list of the Document Titles offered might be

SPRUCE	ELM	MAGNOLIA	ROSE	LAUREL
OAK	HICKORY	TULIP	HOLLY	
DOGWOOD	PINE	DAISY	LUGUSTRUM	

Each of the Titles above would be assigned to its Document at definition time.

Keywords

The contents of a Document may suggest several words that are reflective of its subject material. These **Keywords** provide access to a Document on a generic or topical basis.

A particular Keyword can apply to any number of different Documents, and a Document can have several Keywords associated with it. A Keyword request retrieves all the Documents associated with that Keyword.

Keywords can be up to 15 characters in length, alphabetic or numeric,but they cannot contain periods (or any other special characters, for that matter).

The Titles of the Documents in the sample database mentioned above lend themselves well to the assignment of several Keywords. Some of them might include

Keyword	**Title**
TREES	SPRUCE
	OAK
	DOGWOOD
	ELM
	HICKORY
	PINE
	MAGNOLIA
CONIFERS	SPRUCE
	PINE
HARDWOODS	OAK
	HICKORY
FLOWERS	MAGNOLIA
	DOGWOOD
	TULIP
	DAISY
	ROSE
SHRUBS	ROSE
	HOLLY
	LUGUSTRUM
	LAUREL

Keywords are assigned to a Document at definition time. You can also change a Document's Keywords during normal maintenance processing.

Synonyms

Yet another level of access associates Keywords with other words of similar meaning, or **Synonyms.** Keywords and Synonyms function identically, but as a feature they offer an accommodating flexibility to User requests.

A Keyword can have several Synonyms, but a Synonym can refer to only one Keyword.

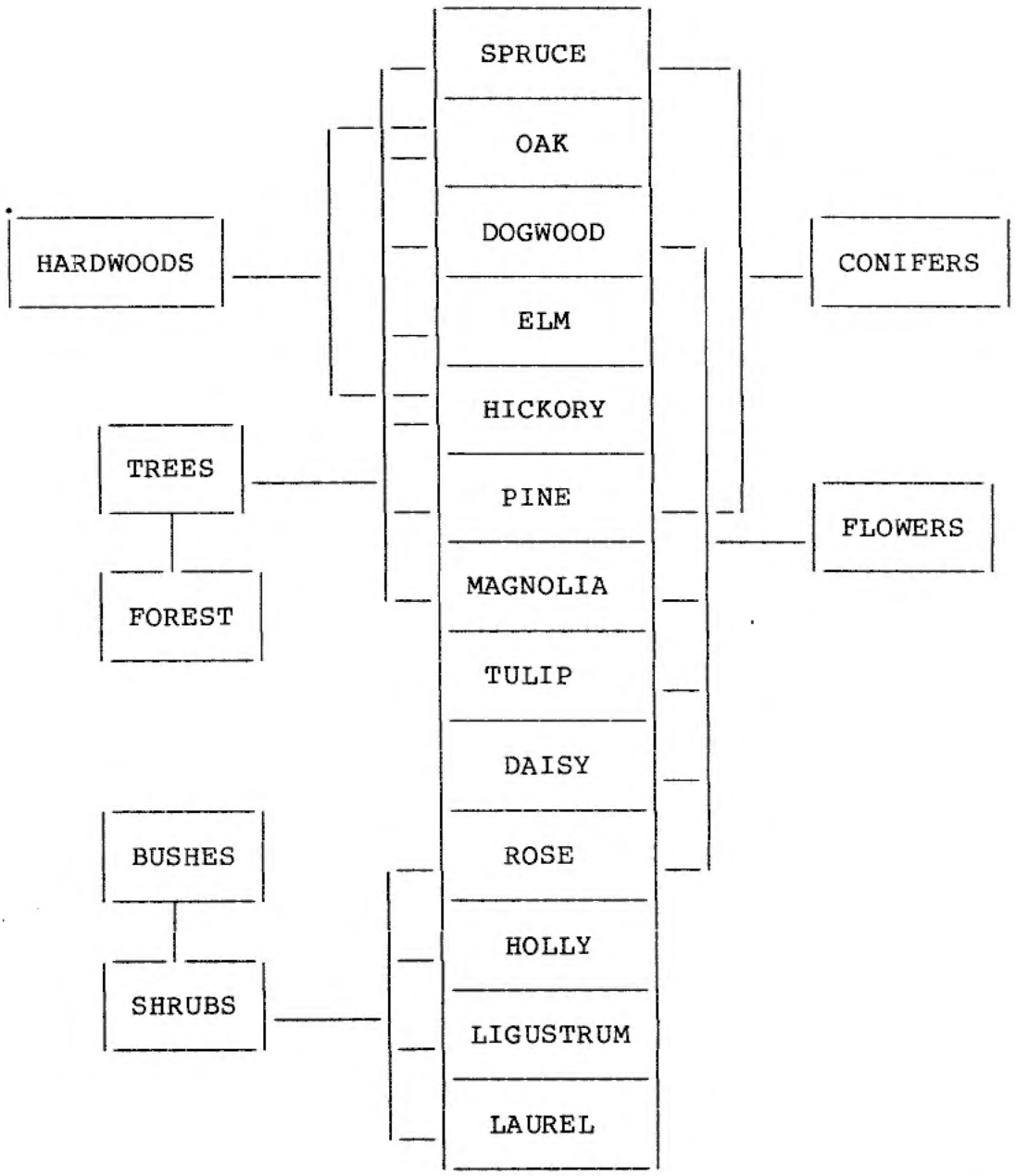
Synonyms, like Keywords, can be 1 to 15 characters in length, consisting of letters and/or numbers, and cannot contain periods or any other special characters.

In keeping with our botanical database example, some of the Keywords shown above could be assigned Synonyms to broaden the access path to the various groups of related articles:

Synonym	**Keyword**
FOREST	TREES
BUSHES	SHRUBS

Unlike Titles and Keywords, Synonyms are not defined when the Document itself is defined. Instead, Synonyms are tied to their Keywords via special database entries called Synonym Documents. These Documents are added and deleted in the same manner as are the other VIS Documents. Maintenance procedures are discussed later in this guide.

The following diagram shows our sample set of Documents and the relationships between its Titles, Keywords, and Synonyms:



Sample Database Title, Keyword and Synonym Relationships

Spelling Variance vs Exact Context

Written language, particularly English, presents many obstacles in the path to its mastery. Spelling alone is such a troublesome endeavor that it finds its way into educational curricula even as far as university level. Most of the idiosyncrasies of English lexicography need not interfere with a User's retrieval session. VIS allows for a wide range of **Spelling Variances** as part of its normal Document access and retrieval processing.

Phonetic Evaluation

A User's word request, if not exactly matched with a Topical Reference in the VIS directory, is systematically compared with alphabetically similar words. Each of these words is rated according to how closely it matches the requested word in letter arrangement (vowels and consonants, number of letters, positional correspondence, etc). This process is known as **Phonetic Evaluation.** The Topical Reference that most closely matches the requested word is then processed normally. Phonetic Evaluation is the mechanism behind the Spelling Variance feature.

The Phonetic Evaluation process is guided by several pre-set internal variables. These values can, as dictated by system performance and User response, be modified. The System Operator should be consulted if the process needs to be tuned. However, the variable settings in effect have been carefully selected to represent the "optimum" values for normal VIS processing and will probably never require modification.

Spelling Variance

The Phonetic Evaluation process provides a vehicle by which VIS can accommodate the following potential "rough spots" encountered in everyday written language and the Spelling Variants they tend to produce during VIS use:

Singular and Plural Word Forms
Verb Tenses
"Accepted" Word Variations
Troublesome Spellings
Typographical Errors

Words like TREE, FLOWER, SHRUB, and BUSH could as effectively be expressed in their plural forms at request time. Phonetic Evaluation should have no difficulty matching these Spelling Variants to their plural counterparts in the VIS directory:

Keyword	Plural Variant
TREE	TREES
FLOWER	FLOWERS
SHRUB	SHRUBS
BUSH	BUSHES

Similarly, verb tenses, as in the words RECEIVE, RECEIVED, RECEIVING, or RECEIVES, should pose little problem for Phonetic Evaluation. Specifying any of these word forms should lead to the one that actually resides in the VIS directory:

Keyword	Verb Tense Variants
RECEIVE	RECEIVES
	RECEIVED
	RECEIVING

Many words exhibit different spellings as "legitimate" or "accepted" variations. COLOUR for COLOR, GELATIN for GELATINE, TYRE for TIRE, and CRAPE MYRTLE for CREPE MYRTLE are examples of such variants. These are also quite acceptable within the context of Spelling Variance and could be used with comparable effectiveness as request words:

Keyword	"Accepted" Variant
COLOR	COLOUR
GELATIN	GELATINE
TIRE	TYRE
CRAPEMYRTLE	CREPEMYRTLE

Some particularly troublesome words frequently undergo spontaneous mutations in the absence of a dictionary. MISSISSIPPI and TENNESSEE are two good examples. A missing consonant or a substituted vowel (as in MISSISIPPI or TENNISSEE, for instance) would not be sufficient in most cases to inhibit a User's request:

Keyword	Misspelled Variant
MISSISSIPPI	MISSISIPI
	MISISSIPI
	MISISIPI
TENNESSEE	TENNESE
	TENESSEE
	TENESEE

Most of the Spelling Variants processed by VIS are simple spelling mistakes or keystroke mishaps. Such casual spelling errors as SCRUB for SHRUB and FLIWER for FLOWER may also (provided there are no SCRUB or FLIWER Documents in the database) pass the Phonetic Evaluation process as acceptable Spelling Variants:

Keyword	Typographical Variants
SHRUB	SCRUB
	SHRIB
	SHRUBB
FLOWER	FLIWER
	FLOUER
	FLOWERR

Word Differences

Variants such as FLOUR for FLOWER or RELIEVE for RECEIVE, however, would probably not make it (as Spelling Variants) through Phonetic Evaluation to their intended references. These are obviously different words, not variations of the same word. "No Information" would greet the User, or some other Topical Reference would emerge as "best" match for the unmatched request word and deliver seemingly inappropriate or unexpected Documents.

Keyword	Different Word
FLOWER	FLOUR
RECEIVE	RELIEVE
HARDWOOD	HARDWARE

Exact Context Requests

The User who finds himself viewing unexpected material in response to a request should

1. try typing the same reference again to eliminate the possibility of a casual typographical error, or

2. request that VIS process the reference in its **Exact Context** by enclosing the request in quotes (" "), which in effect suppresses Phonetic Evaluation.

An Exact Context request insists that only an exact match for a specified reference is acceptable as a response. If "No Information" is returned, then it is obvious that the reference used is useless. The User should then

3. try another word as a Topical Reference in an attempt to approach his subject material from a different angle.

For example, a User may have entered BRUSH as a request and received the BUSH Documents at his terminal instead of information about the latest in home cleaning aids. To determine absolutely whether there are any BRUSH Documents in the VIS database, he can enclose the word in quotes ("BRUSH"). Assuming that there are no BRUSHES, the User would get "No Information". He can then try a different Topical Reference, say CLEANING.

Spelling Variance is not a cure...

Spelling Variance is not a cure for chronic bad typists or habitually poor spellers. Nor does it attempt to interpret or evaluate the "intentions" of a User request. It is, however, a distinctively user-friendly feature that smoothes out some of the anticipated stumbling blocks that language often presents.

The Spelling Variance feature in conjunction with the Topical Reference access mechanism provide a liberal yet disciplined environment for User interaction. The rigid constraints of most computer interface vocabularies are replaced in VIS with an almost human willingness to accommodate any reasonable (and some not so reasonable) User requests. This quality is inviting to novice and experienced Users alike.

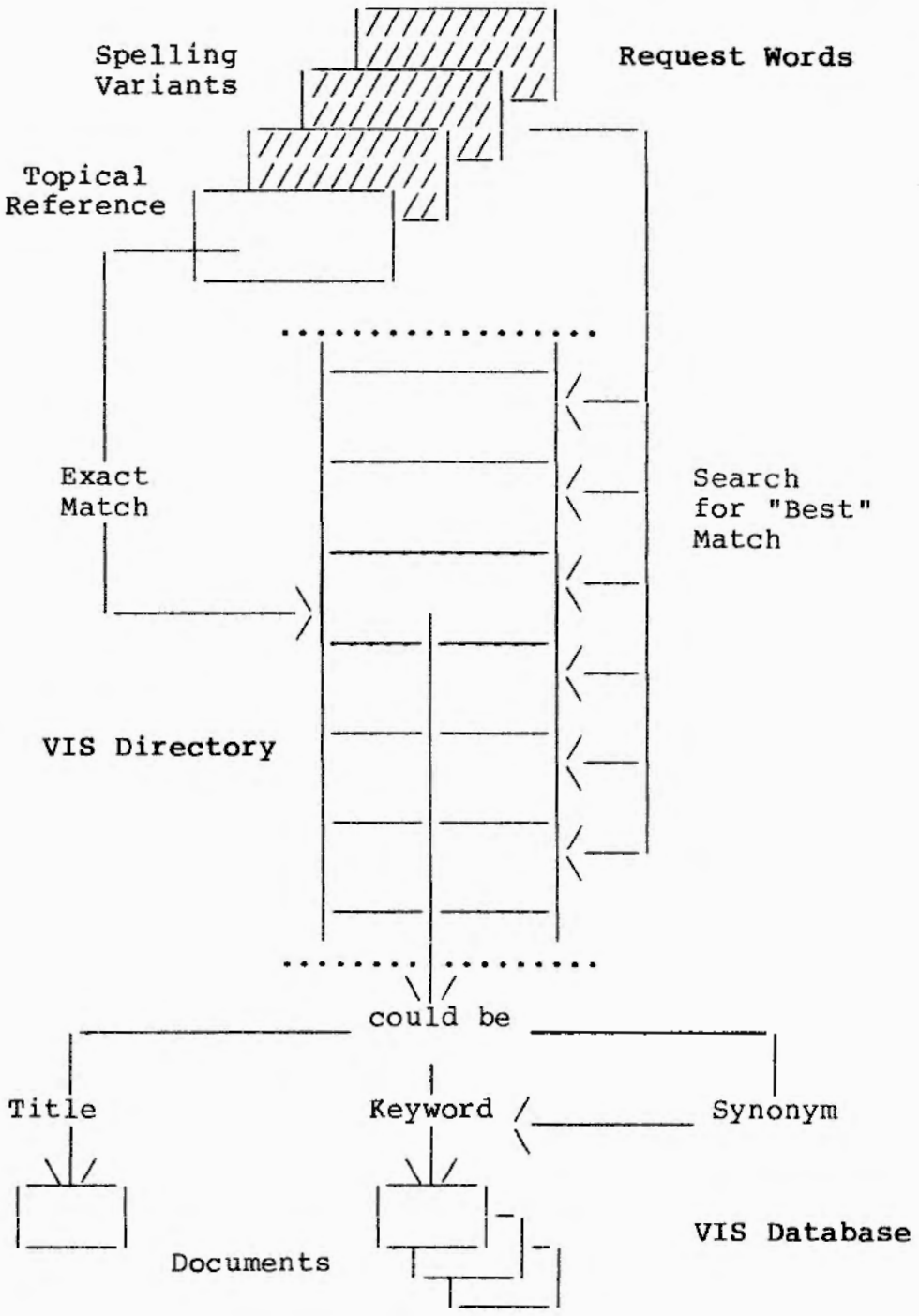
Novices tend to go through a period of discovery and experimentation with respect to a new system in an effort to find out what areas of the database are of the most interest to them and how to get at these items. VIS encourages exploration by rewarding the adventurous beginner with meaningful responses to his perhaps timid initial requests.

Most of the gross errors that Users make occur during the "exploration" stage. Assuming that most Users will learn from their "mistakes", instances of their receiving unexpected or unwanted data should decrease dramatically with experience and familiarity with VIS. After this learning phase, Users tend to settle into patterns of "normal" use.

However, even after novelty yields to habit Users will continue to find the "shortcuts" that the flexibility of the Spelling Variance feature and the power and versatility of Topical Reference processing tend to nurture.

Experienced Users are always improving their VIS sessions by experimenting with new words and combinations, discovering alternate reference paths to their favorite information items, and finding different relationships between Documents. VIS access methods and request processing promote this type of activity, although the extent and effectiveness of such activity is directly dependent on your own inventiveness and sensitivity.

The following diagram illustrates how VIS locates Documents using request words (Topical References and Spelling Variants) submitted by a User:



VIS Request Processing

Multiple Reference Requests

User requests are not limited to single word entries. In fact some of the most powerful capabilities of VIS Document retrieval arise only in the context of **Multiple Reference Requests.**

VIS accepts several words in a single User request, and processes them one at a time as they are encountered in the input line. A Title causes a particular Document to be retrieved. A Topical Reference (Keyword or Synonym) typically produces a list of related Documents.

The lists of Documents resulting from Multiple Reference Requests can be selectively tailored by "logically combining" them. Two or more Topical References joined by a Logical Operator character produces a **Logical Combination.**

The **Logical Operator** characters provided in VIS request processing are:

, (comma)	Logical OR
/ (slash)	Logical AND
: (colon)	Exclusive OR (XOR)
- (dash)	Logical Subtraction

The facility to process Multiple Reference Requests and selectively combine them offers great flexibility in tailoring information to suit virtually any need. A carefully devised reference structure can make this feature a powerful tool: a minimum effort will render it feeble - it's up to you.

ORing Multiple References

In many cases, the lists generated by two or more separate references can involve the same Documents. Such overlaps need not yield redundant material at the User end.

OR (,) specifies that the lists of Documents produced by the Topical References separated by the , are to be combined into a single list containing all the Documents in BOTH lists WITHOUT DUPLICATING any Documents that appear in both.

Recalling our sample database, the two Keywords TREES and FLOWERS could be ORed as follows:

TREES	FLOWERS	TREES,FLOWERS
SPRUCE		SPRUCE
OAK		OAK
DOGWOOD	DOGWOOD	DOGWOOD
ELM		ELM
HICKORY		HICKORY
PINE		PINE
MAGNOLIA	MAGNOLIA	MAGNOLIA
	TULIP	TULIP
	DAISY	DAISY
	ROSE	ROSE

ANDing Multiple References

It is also useful to be able to select Documents by virtue of their inclusion in more than one topical category. This permits the User to target his requests at items satisfying any number of specific subject requirements he wishes to impose.

AND (/) specifies that the lists of Documents produced by the Topical References joined by the / are to be combined into a single list containing ONLY THOSE Documents that appear IN BOTH the original lists:

TREES	FLOWERS	TREES/FLOWERS
SPRUCE		
OAK		
DOGWOOD	DOGWOOD	DOGWOOD
ELM		
HICKORY		
PINE		
MAGNOLIA	MAGNOLIA	MAGNOLIA
	TULIP	
	DAISY	
	ROSE	

EXclusive ORing Multiple References

On the other hand, a User may NOT want to see certain Documents among a related group. As long as the unwanted items can be sub-topically related (i.e. the Documents in question share a Topical Reference), excluding them is a simple matter.

XOR (:) specifies that the lists of Documents produced by the Topical References joined by the : are to be combined into a single list containing BOTH sets of Documents EXCLUDING THOSE that appear IN BOTH lists:

TREES	FLOWERS	TREES:FLOWERS
SPRUCE		SPRUCE
OAK		OAK
DOGWOOD	DOGWOOD	
ELM		ELM
HICKORY		HICKORY
PINE		PINE
MAGNOLIA	MAGNOLIA	
	TULIP	TULIP
	DAISY	DAISY
	ROSE	ROSE

SUBTRACTing Multiple References

A Topical Reference may deliver items that a User considers superfluous. As long as these items are topically related by a different Keyword or Synonym they can be excluded from the larger group using this Logical Operator.

SUBTRACT specifies that the list of Documents produced by the Topical References joined by the - are to be combined into a single list containing all of the Documents in the first list EXCEPT those in the second:

TREES	FLOWERS	TREES-FLOWERS
SPRUCE		SPRUCE
OAK		OAK
DOGWOOD	DOGWOCD	
ELM		ELM
HICKORY		HICKORY
PINE		PINE
MAGNOLIA	MAGNOLIA	
	TULIP	
	DAISY	
	ROSE	

Text Programmed Presentation (TPP)

It is the Text Programmed Presentation feature that helps to distinguish VIS among existing Videotex systems. It adds the dynamic and artistic potentials traditionally absent in the domain of ordinary information storage and delivery.

Though Text Programmed Presentation represents a powerful addition to the tools used in information management, the simplicity and flexibility of its application put its mastery well within the abilities of non-technical copy editing or word processing personnel. It turns the otherwise lack-lustre activity of data management into a challenging and even creative endeavor.

Text Programmed Presentation, TPP for short, consists of special character sequences included in the bodies of Text Documents. These sequences direct VIS in the formatting, presentation sequence, and interactive operations of your Documents. VIS recognizes these sequences while Documents are being prepared for transmission, and processes them accordingly at that time.

TPP Character

The TPP Character triggers VIS TPP processing, and is immediately followed by a TPP Code. Since this character can occur anywhere in a Text Document, it should be as "special" a character as possible so as to avoid potential conflicts with your normal text characters. The TPP Character can be set to the value that is most compatible with the contents of your Documents. (Consult your System Operator if you feel the need to do so.) The default character is * (asterisk).

TPP Codes

There are eight TPP Codes currently available for use in Text Programmed Presentation: four of them affect only the Text Document in which they occur and are generally concerned with Document formatting, the rest direct Document processing in special ways and require Titles, parameters or other arguments. TPP Codes and any associated parameters and values follow the TPP Character in the Text Document.

Use of the various TPP Codes is discussed below:

***END**

This TPP Code signals "end of text" to VIS. It requires no additional parameters, although it must be the last thing specified in the Text Document file.

On encountering the *END code, VIS transmits the screen and either waits for another request from the User or proceeds to process the next Document in the request queue.

The following example shows how the *END code is included in the body of a Text Document file:

```
...U.S. Navy today referenced its new nuclear
powered ship contract...at least 30 of the ships...
Mediterranean fleet.
*END
```

*PAGE

Terminal screens exhibit a wide variety of screen capacities and formats, expressed in terms of "characters per line", "lines per screen", and their product "characters per screen". At transmission time VIS sends out a "screen" of characters based on a terminal's defined capacity.

You can override this "page" formatting to improve the readability and aesthetic quality of your Documents. For this reason, you should be aware of the "typical" screen formats that are to display your Documents.

Here are some common screen configurations:

MODEL	LINES PER SCREEN	CHARACTERS PER LINE	CHARACTERS PER SCREEN
DT-1	24	80	1920
Video.Term.	16	32	512
Color Cptr	16	32	512
I	16	64	1024
III	16	64	1024
II	24	80	1920
4	24	80	1920
12	24	80	1920
16	24	80	1920
100	8	40	320
PT-210	-	80	-
Tandy 2000	24	80	1920

Large bodies of text are harder to read when allowed to "scroll" as one continuous stream on a User terminal. In addition, many items of text have "logical" breaking points that can likewise suffer a loss of readability and visual impact when presented in an unformatted way.

The *PAGE code allows a large or segmented body of text to be broken up into separate screens. It can occur anywhere in the text. Like END, no additional arguments are required, although the code must be followed by either a space, a tab character or a <CR> (carriage return). VIS writes out the prepared screen. Before continuing to the next page, VIS either waits for the User to make some response, or pauses for the interval of time set by the PAUSE= specification during User definition.

Consider this sample Document body:

```
The following material is recommended for mature
readers ONLY!
*PAGE
The problem of sex discrimination in ...alarming...
men and women...for centuries!
*PAGE
Children's efforts to discover...Bureau of Land
Management...many cases of outrageous public
exhibitionism....
*END
```

*CLS

Screen "clutter" is also an inherent problem when dealing with continuous streams of textual data. Left-over material from one request rarely adds to the readability of the new material of the next. The need for a "clear" screen is even more critical if graphics are to be displayed on the terminal.

The CLS code causes the User's terminal screen to be cleared before new information is written to it. CLS must be followed by a space, a tab character, or a <CR>.

This code will usually occur at the beginning of a page of text:

```
*CLS
Here are today's major news stories:

The IRS reported today that all income taxes will
be suspended until further notice...
*PAGE
*CLS
Lake Superior has mysteriously dried up for the
third time this month...
*END
```

*REM (remarks)

It is sometimes useful to include comments or remarks along with text that do not get written to the User terminal. As well as being a good "documentation" aid, the REM facility can be used to provide directions or instructions for others who might be involved in preparing or revising Documents on a regular basis.

A REM can appear anywhere in the text body, followed by a space and the desired comments enclosed in parentheses. Each remark can be up to 512 characters long.

You might use the REM code like this:

```
*CLS
...our new articles on the following topics:

*REM (Change this list according to the new items
shown on Cheryl's blackboard)
     Dog Grooming
     Home Improvement
     New Tax Laws
     Famous Foreign Podiatrists
*PAGE
And for our younger Users:

*REM (Don't change these - they're still the same)
     Cartoon Trivia Quiz
     Biographies of Famous Scientists
     Video Game News
*REM (Tell Jack when you're done so he can get this
into the next transmission)
*END
```

*INCL ("Title")

There are times when the ability to "nest" Documents becomes particularly useful. The most obvious example is in the presentation of text in conjunction with graphics: a map of the state with temperatures positioned to correspond to selected cities; a color bar-graph superimposed with explanatory text; a particular part diagram with component description and part number.

The INCL code provides the mechanism for this useful facility. The code is followed by a space and a Title (in quotes) enclosed in parentheses. The Title can be up to 26 characters in length, and you must have already defined the specified Document to VIS. VIS begins processing the named Document immediately on encountering an INCL code, and then returns to complete the Document in which the code was encountered.

Here is a typical example of how the INCL code might be used:

```
*CLS
*INCL ("D0115HT") *REM (Diagram of 0115HT)

    Front Panel


                                        Lock Nut


                    Ratchet Pin
                    Assembly


    Part # 0115HT fits all models except those designed
    especially for export to Argentina. Made of durable
    stainless steel (plastic Front Panel). Butter Churner
    Association rating of 00.45.
    *END
```

In this example, "D0115HT" is the Title of a Graphic Document, already defined, that is transmitted to the terminal screen first, then superimposed by the descriptive text.

***MENU (integer"Title" ... ELSE"Title")**

It is very useful to be able to "guide" a User through a complex selection process. "Large" subjects frequently need to be narrowed down in scope to isolate particular areas of interest, and these areas of interest likewise need to be broken down into component Documents for access and retrieval. The MENU code provides this extremely useful capability.

A screen of text can show a list of "choices" alongside a list of corresponding numbers. Each number is associated with a Document Title (in quotes). The list of choice numbers and their Titles is contained within parentheses and follows the MENU code and its adjacent space. Each individual choice number and Title is separated by a space. VIS waits for a response and, according to the value returned by the User, processes the appropriate Document.

The special ELSE clause points to an "invalid response" Document. If one is not coded among the MENU choices and the User enters an invalid response, VIS will reprompt for a correct response. The number of reprompts is the value of the RETRIES= specification in the User Detail entry.

Let's assume that our "0115HT" Document, discussed in connection with the INCL code to illustrate how a diagram might be INCLuded in a Text Document presentation, is one of several part description Documents for one of several small appliances managed by a particular inventory system. Assume that all the part description Documents have already been defined. A Menu, call it "M01", could be set up something like this:

```
                MIRACLE BUTTER CHURNER

                     1  Base
                     2  Container
                     3  Hand Driver Mechanism
                     4  Churning "arms"


*MENU (1"0117ZX" 2"0113AW" 3"0115HT" 4"0121CA"
       ELSE"INVRESP")
*END
```

Each numbered choice points to a Document Title. A User's response is matched to one of the numbers (or ELSE), and the Document is processed.

Now that this Menu is available for use, a "higher" level selection can be set up that leads to the above Document (and other such "sub-Menus"):

```
                HOME MAKER HELPERS

                1  Miracle Butter Churner
                2  Speedy Meat Tenderizer
                3  Jumbo Juicer
                4  Mamouth Mixer
                5  Two-Ton Toaster


     *MENU (1"M01" 2"M26" 3"M74" 4"M31" 5"M85"
            ELSE"INVRESP")
     *END
```

This Menu might be called "HELPERS" and it could be given an assortment of Keywords to allow flexible User access, such as HOME, HOMEMAKING, KITCHEN, etc. A User enters his choice number, say 1, and "M01" appears, 2 gets him "M26", and so on.

A useful addition to a Menu text would be a reminder of the various <BREAK> options available to a user. This would be especially helpful to the neophyte VIS user who is still not fully acquainted with all of the VIS commands. Using the previous example, the <BREAK> reminders might be integrated like this:

```
                HOME MAKER HELPERS

                1  Miracle Butter Churner
                2  Speedy Meat Tenderizer
                3  Jumbo Juicer
                4  Mamouth Mixer
                5  Two-Ton Toaster

     Hit 'd' then <BREAK> to skip to next document.
         'w' then <BREAK> to skip to next request.
         <BREAK> to exit menu selection.

     *MENU (1"M01" 2"M26" 3"M74" 4"M31" 5"M85"
            ELSE"INVRESP")
     *END
```

Alternatively, the additional three lines of text might be stored as a separate document and INCLuded as you see fit.

***WHEN (argument"Title" ... ELSE"Title")**

Some terminals do not produce graphic images. Some do not have the capability of receiving Programs. In these situations it is useful to be able to direct the sequence of Document presentation without eliciting a User response. The WHEN code provides this capability.

Following the WHEN code are two intervening spaces. These two spaces are required. After the two spaces, is a list of paired arguments or values and their Document Titles (in quotes). This list is enclosed in parentheses and the items are separated by spaces. Each argument specifies a terminal type specification (one of the three-digit TERMINAL= values, or one of the "standard" code designations) that, if matched to tne value stored by VIS at Logon time, directs VIS to the appropriate Document. If not, the ELSE Document is retrieved.

Assume that terminal type 008 identifies a device that is capable of producing "high-res" graphic images, and type 016 terminals cannot. Instead of sending a "Wrong Terminal Type for Graphics" message to a type 016 User, an alternate Document, one that employs text rather than graphics, can be sent. Our Miracle Butter Churner, for instance, could be described in two sets of Documents and, according to the User's terminal type, the appropriate Documents would be sent at request time. The "0115HT" Document Title specified in the *MENU code of "M01" could point to a Text Document like this one:

```
...
*TEXT
*WHEN  (008"A0115HT" 016"B0115HT" ELSE"B0115HT")
*END
```

"A0115HT" would point to our original graphics-producing example, and "B0115HT" to a text-only version. Note the ELSE case points to the text-only version also, just to cover the bases. Incidentally, the contents of this little directing Document would probably not occupy its own file, but would be included as part of the Verb itself, as the *TEXT...*END construction implies.

***LOAD ("program name""arguments")**

It is sometimes useful to be able to process or interpret User responses in a more detailed manner than is provided by the MENU code. Programs to accomplish processes such as these can be stored in the VIS Host file system and invoked via the *LOAD code. These programs can also serve as a means to allow Users to indirectly interface with TRS-XENIX.

Programs invoked by the *LOAD code should not be confused with Program Documents. With Program Documents, a User either interacts directly with the program ("arcade" mode) or has the program downloaded to his terminal ("disk transfer" mode). A User's interaction with programs invoked with the *LOAD code is completely indirect. A one word response from the User is passed to the program, and this response is then processed by the program. The program can then return responses either directly to the User or as commands to VIS that will then be displayed to the User.

The *LOAD code is followed by a space, then (in parentheses) the program name (in quotes). Any arguments or values (also in quotes) to be passed to the LOADed program are optional.

In the following example, a Text Document asks the User to confirm his desire to view a particularly expensive set of Documents:

```
...
The Documents you have requested will cost
$15.00 to view due to the natural law of
supply and demand.

Please confirm your request by typing "Y" in
response to the LOAD: prompt.
*LOAD ("TEST""FRUITS HIMOM")
*END
```

When VIS encounters the *LOAD code in a Text Document the program named in the first set of quotes is invoked. The arguments (if any) in the second set of quotes are available to the program as if they had been typed as part of a command line from a TRS-XENIX terminal. The User response, one word only (a word is delimited by a blank, <ENTER>, ';', or null character), is also available to the program via standard input commands. Output is written using standard output operations.

The program TEST (note that this name must be in upper case letters) is written in "C". Once the program has been written, it is compiled and linked just like any other "C" program, and it is subject to the same considerations of other programs. It is invoked as its own task of Xenix and therefore some thought should be given to the management of system resources. A large program will require a commensurate amount of system resources, perhaps significantly impacting on the concurrent processing of VIS.

Once the program is in executable form, it is stored under the name that appears in the *LOAD sequence, e.g. TEST, in your **/custprog** directory (see "The VIS Customer File Directories" in the next section).

When a User response has been processed by the program, your program can elect to submit a command directly to VIS by prefixing the request line with a lower case "c", or simply issue a message back to the User terminal by prefixing the message with a lower case "s". If neither of these prefixes are used, the current default is to treat the output as a command.

In the example program TEST, the User response is examined and, if it is a "Y", the command "HIMOM ROSE FRUITS HIMOM" is sent directly to VIS, which will then display the appropriate documents to the User. If any other response is given by the User, the "test LOAD ESCAPE" message is sent to the User terminal and the Document exits normally. The source code for this sample "C" program can be seen on the following page:

```
#include <stdio.h>
main(argc, argv)

int argc;       /* argc is number of arguments */
char *argv[];   /* argv is an array of pointers to
                           the arguments */
{

     char buffer[20];     /* array used to hold User
                              input and scanf reads in
                              the first or only word */
     short i;

     scanf("%s", buffer);
     if (strcmp(buffer, "Y") != 0)
     {
          /* the first character written out is 's' to
               indicate that a null-terminated string
               is to be displayed by VIS */
          putchar('s');

          printf("This is a test of the LOAD
                  ESCAPE\nString Version !!!\n");
     }
     else
     {
          /* the first character written out is 'c' to
               indicate that a command to be submitted
               to VIS follows */
          putchar('c');
          printf("himom rose %s", argv[1]);
     }
}
```

TPP Code Summary

***END**
marks "end of text"

***PAGE**
"end of page", pause

***CLS**
clear terminal screen

***REM (remarks)**
remarks (not sent)

***INCL ("Title")**
include Document in presentation sequence

***MENU (integer"Title" ... ELSE"Title")**
match User response to an "integer" (or ELSE)
and process Document

***WHEN (argument"Title" ... ELSE"Title")**
match TERMINAL= value with "argument" (or ELSE)
and process Document

***LOAD ("program name""arguments")**
load program, pass "arguments", route a User
response to program

Preediting Program Documents

As mentioned earlier, Program Documents must be stored in the CompuServe B Protocol format to accommodate their eventual transmission. The utility to accomplish this is **/vis/pre_edit.** To use pre_edit, type:

```
/vis/pre_edit arg1 arg2 arg3 arg4  <inputfile  >outfile
```

where:

arg1 -- Must give this argument. It can be either -DT (disk transfer) or -MT (memory transfer)

if -DT is the option, then:

arg2 = filename : This will be name of the file once it is transferred on to user computer's disk.

arg3 = 'B' (binary file download) or 'A' (ascii file download)

if -MT is the option, then:

arg2 = load address of the program (in hex)
arg3 = transfer address of the program (in hex)

arg4 : Optional. This is the timeout value in sec, during the protocol (in dec). The default value is 20 sec and this is also minimum value.

inputfile : File which you want to pre_edit.

outfile : Name of the file which will be put into the database.

examples:

```
/vis/pre_edit -MT 2ab3 2ad9  <cube.abs  >cube
```

will cause cube.abs to be pre_edited for a memory transfer with 2ab3 being the load address and 2ad9 being the transfer address. Since arg4 was missing, the default value of 20 sec is used for timeout.

```
/vis/pre_edit -DT CUBE/CMD B 30  <cube.bin  >cube
```

cube.bin is pre_edited for a disk transfer which when transferred will have the name CUBE/CMD. It will be in binary form. The timeout value now is 30 sec. cube is the outfile which is going to be inserted into the database.

```
/vis/pre_edit -DT PLANT A  15  <plant.txt  >plant
```

plant.txt is pre_edited for a disk transfer which when transferred will have the name PLANT. It will be in ascii form. plant is the outfile which is going to be inserted into the database. Since the timeout value given is 15 sec, which is less than the minimum value of 20 sec, the 20 sec value is used for the timeout value.

Maintaining the VIS Database

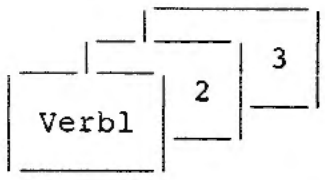
A tandem process has been developed to perform VIS database maintenance. It is called **Edit/Maintain.** The Edit/Maintain process is set up to run at periodic intervals, and it is initiated by the VIS System Operator. All the input you submit after an Edit/Maintain run but before the next scheduled execution will be processed. Any corrections made to faulty input of one run are submitted to a subsequent run along with any new input.

The input to Edit/Maintain is made up of two basic types of information: Verbs and Document contents. These "raw materials' are prepared and submitted by you for editing and inclusion into the VIS database.

Edit/Maintain processes your raw materials (Verbs and Document contents) in a two-stage procedure. Each Verb is first edited for correctness of syntax and interconsistency of specification and merged with any corresponding Document contents into a partially processed form. This first step is called **Edit.** The pre-processed material produced by Edit is then used to perform the actual VIS database updates. This second step is known as **Maintain.** Pre-processing maintenance materials in one stage and updating the VIS database in another reduces the time during which User access to the Documents being changed or added is inhibited.

Verbs

A **Verb** is a set of instructions, specifications and values that direct the VIS Edit/Maintain processor to add, change, delete, list or report on User data and Documents in the VIS database. Each Verb requests a specific maintenance function with respect to a particular database entry. Verbs can be submitted separately as individual files or they can be grouped into a single multi-Verb file:



Separate Verb Files

Multi-Verb File

Document Files

The contents of most Documents are maintained in **Document files.** For Program and Graphic Documents, this is mandatory. Text Documents are usually prepared as separate files, but they can also be submitted as part of their Verb specifications using the *TEXT feature (more later). Input Documents share this option, but due to the generally small size of the request lines they contain, Input Documents are usually submitted with their Verbs specifications. Note that User Master and User Detail Documents have no contents other than the specifications and values contained in their Verbs.

Preparing Verbs and Document Contents

The preparation of raw materials (Verbs and Document files) for processing by VIS Edit/Maintain can be accomplished in one of two ways, depending on the particular environment in which you operate as VIS Customer and the arrangements you make with the VIS System Operator.

Note that regardless of the specific method you use, the contents of your prepared materials are the same. Though the discussions that follow emphasize the contents of your maintenance materials, they also include references to specific techniques and commands applicable to one or the other of the two basic modes of data preparation and manipulation.

One other item to keep in mind when preparing text documents is that all of the non-printable characters in the ASCII character set are ignored as input to VIS, except those for tabs, carriage return, line feed, and space. Such things as boldface, underlining, etc., will not show up in the resulting VIS document, although they will not be regarded as errors if they are left in. The reason for this restriction is because VIS is accessible to many different terminal types, of which some would interpret these characters in a non-standard, unpredictable fashion.

TRS-XENIX Text Editors

Among the many facilities of the TRS-XENIX product are a number of text editors and file manipulation commands that can be used for managing your VIS materials.

You would use the TRS-XENIX text editing facilities exclusively if:

> You are functioning as both System Operator and Customer in a "small" service environment (in which case the system console serves as system communication device and VIS work station).
>
> You are an "on-site" Customer using a Model 12 or other terminal device as a separate work station connected directly to the Host Model 16.
>
> You are "remote" but connected to the Host Model 16 through a "dedicated" (leased) data line instead of a "dial up" line .

The TRS-XENIX text editors are a large subject to cover in a guide of this scope. Many other facilities available to TRS-XENIX users in general would also enter into such a discussion. For these reasons it is recommended that if you intend to make extensive use of the TRS-XENIX facilities, consult the abundant documentation dedicated to these and other appropriate subjects.

Scripsit or Other Word Processor

This is the "preferred" medium for preparing VIS maintenance materials. Several major advantages are inherent in this method, and they apply to both "small" and "large" services:

> Your equipment functions in all areas of your enterprise, not just in VIS interface operations. Accounting, payroll, general word processing, financial planning, educational aids, entertainment and many other applications can make use of the same resources you use in your job as VIS Customer.
>
> Your service remains independent of the VIS Host, interfacing only when necessary. Contact with the Host is brief, "connect" time is short and inexpensive.
>
> Of particular value if you are "remote" is that the same equipment that you use to maintain your Documents can be used to view these Documents as one of your own Users.

Scripsit is a versatile, full-featured word processing system available for use on most of the TRS-80 microcomputers. Materials prepared with Scripsit are easily configured for transferring or "uploading" to the VIS Host.

Instructions on the use of Scripsit (or any other word processing system you use) are not included here. Emphasis is on the "content" of the materials you prepare, not the techniques and specialized instructions involved in their physical preparation.

Naming Verb and Document Files

The names you select for your Verb and Document files are not entirely free from conventions:

> Verb file names must begin with the prefix **raw.** and the rest of the name is up to you
>
> Document file names must consist of up to the first 14 characters of the Document Title (NAME= specification in the defining Verb) The name must be in lower case and periods are ignored.

There are several points along the route from preparation to processing at which the above conventions can be applied, so it is not necessary that you adhere to the standards from the outset. It is necessary, however, that a Verb or Document file be correctly named prior to the Edit/Maintain run or it will not be processed.

It is recommended that you do follow the file naming conventions throughout your file management activities so that the additional inconvenience of a renaming procedure at some point along the way can be avoided. This point will become clear in the discussions of file conversion and transferring prior to Edit/Maintain.

The names you select for your files (and your Document Titles, for that matter) should be composed of letters, numbers, and periods (dots) only. Special characters (@, #, $, %, etc) should be avoided since they may take on special meanings in different contexts, and cause functional and/or processing errors.

The VIS Customer File Directories

VIS runs as a task of the TRS-XENIX operating system, a powerful multi-tasking, multi-user operating system derived from UNIX for use on the TRS-80 Model 16. All Customers are in fact TRS-XENIX users.

The TRS-XENIX operating system manages files by grouping them within defined hierarchies identified by what are called **directories.** Each file is accessed by specifying a composite of the directories within which it resides and its file name. When a Customer is defined to VIS, a sub-system of file directories is established within the VIS file sub-structure.

In the case of VIS Customer files, a three-level directory designation is necessary to identify the "path" to any specific file within his sub-system: the first level (named **/vis**) points to all VIS sub-directories, the second (**/(ocode)**) is the name of your VIS directory, and the third (either **/verb**, **/document**, **/verb.bad**, **/document.bad**, **/vis.output**, **/report**, **/custprog**, or **/edit.output**) identifies the directory that contains or will contain a specific file.

The VIS maintenance process both expects its raw materials to be in certain file areas (directories) and places its output in specific file areas. When you login to the TRS-XENIX system to prepare and correct your database maintenance materials, you are placed in your "home" directory, e.g. **/vis/(ocode).**

Note that **(ocode)** stands for your VIS Ownership and ID values concatenated into one value. These values are provided by the VIS System Operator.

Your working directories within **/vis/(ocode)** are:

/verb	contains the Verb files that request and direct maintenance
/document	contains the Document bodies for Text, Program, Graphics, and Input Documents
/verb.bad	contains diagnostic output from Edit/Maintain for Verbs found to be in error

/document.bad	contains Document bodies found to be in error during Edit/Maintain
/edit.output	contains partially processed output from Edit phase
/vis.output	contains output from LIST requests
/custprog	contains compiled Customer programs used in *LOAD processing
/report	contains intermediate and final output from REPORT requests

These directories define the environment in which you carry out your database maintenance functions.

The Flow of Data During Edit/Maintain

The **Edit** phase of the VIS maintenance mechanism pre-processes all the Verb files it finds in your Verb directory (**/verb**) that begin with the characters **raw.**. A Verb that passes Edit successfully is, if necessary, merged with its corresponding Document file. Edit uses the first 14 characters of the NAME= specification in the Verb to locate the appropriate Document file in your Document directory (**/document**). The "object" data produced is filed into your Edit Output directory (**/edit.output**). Utility report requests are written to a separate file in the **/report** directory. (The utility reports are discussed later.)

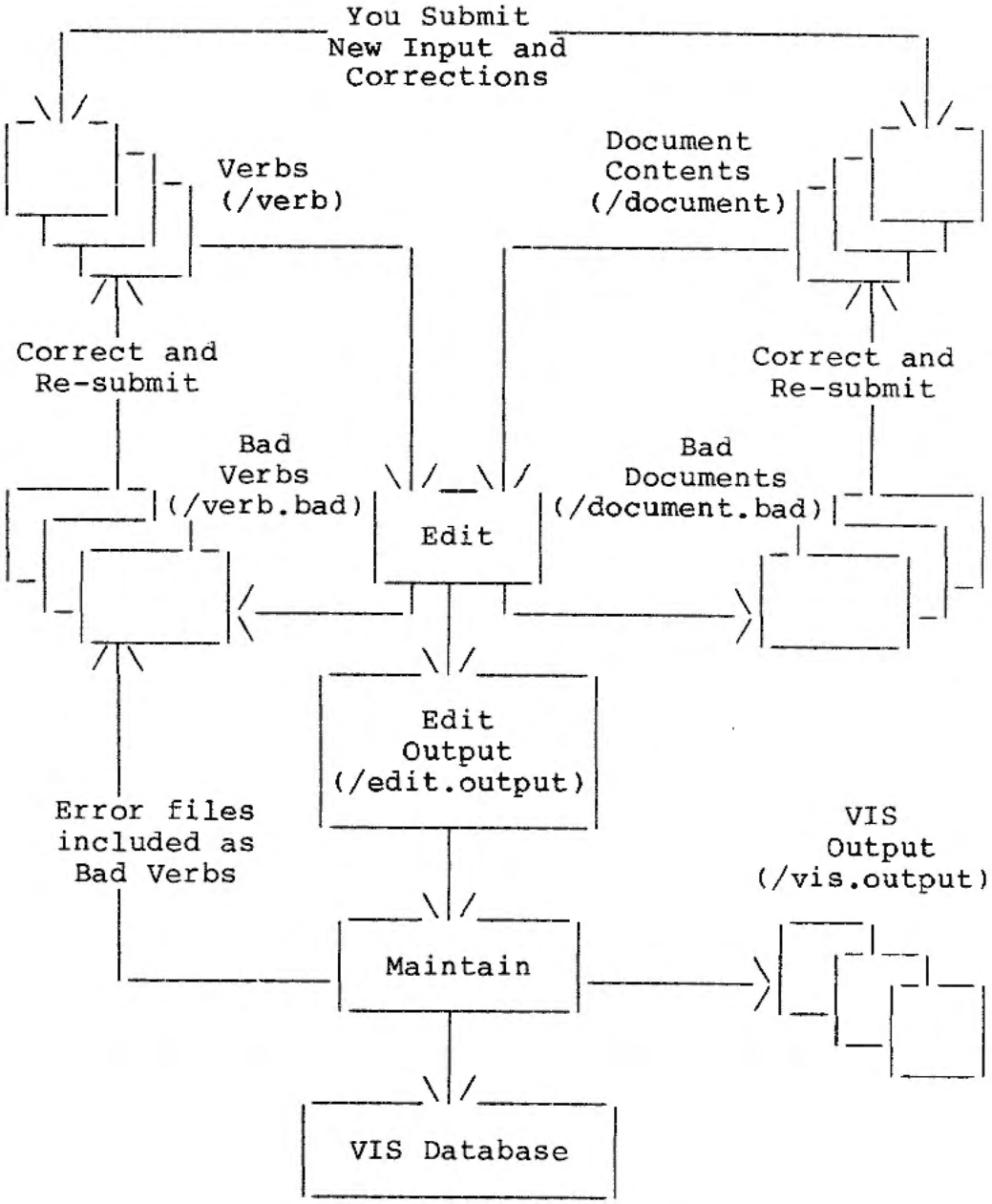
Unsuccessfully edited Verbs and Document files are written to corresponding diagnostic files of the same names, along with the appropriate error messages, in your Bad Verb (**/verb.bad**) and Bad Document (**/document.bad**) directories respectively. Corrections to faulty Verbs and Document files are made using the information in these diagnostic files. Corrected files are then included in the next run's input.

Maintain, the second phase of the VIS maintenance cycle, applies the material contained in the Edit Output file to the VIS database. *LIST output is written to your VIS Output directory (**/vis.output**).

Errors encountered during Maintain cause the faulty Edit Output material to be written to **/verb.bad** under file names of **raw.**+(first 10 characters of NAME= value). Both the Verb specifications and the Document contents (if any), plus error messages, appear in these files. Program and Graphic Documents are exceptions: their contents will appear in the same directory (**/verb.bad**) but with a file name prefix **pg.** instead of **raw.**.

WARNING: All the "raw" files in your Verb directory and your Document directory are purged at the end of Maintain. This is done to avoid processing the same files (good or bad) more than once. Corrected Verb and Document files must be submitted with, and as though they were, new Edit/Maintain input.

The following diagram illustrate the flow of data through your file sub-system (**/vis/(ocode)**) during Edit/Maintain processing:



Flow of Data During Edit/Maintain

Verb Files

Verbs consist of the commands, specifications and values that direct Edit/Maintain in the process of VIS database maintenance. Verbs must follow certain prescribed guidelines with respect to format and content. These syntax requirements are discussed first on a general basis, then in terms of the specific Edit/Maintain commands.

Verbs can be coded in either upper or lower case letters. Although a Verb file can contain more than one Verb, it is recommended that each Verb be coded as a separate file. This is because a Verb found to be in error causes the rest of the file's contents to be ignored; subsequent Verbs within the same file will not be processed.

All Verbs contain three basic elements: an Edit/Maintain **command**, a list of **specifications** and their values, and a Verb **terminator**. The basic scheme of a Verb looks like this:

```
*command
specification=value
     (as many as required)
*terminator
```

The three Verb elements are discussed below.

Edit/Maintain Commands

An Edit/Maintain command is always the first item of a Verb file. It must have an * (asterisk), or whatever the TPP Character happens to be set to (discussed later), as its first character. This character identifies the file as a valid Verb file.

Valid commands are:

***ADD**	add an entry to the VIS database
***MOD**	change the specifications or values of an existing entry in the database
***KILL**	delete the specified database entry
***LIST**	display the contents of the specified database entry
***REPORT**	produce one of the utility reports (discussed later)

***ADD**

The *ADD command instructs Edit/Maintain to incorporate new information into the VIS database.

The data that is ADDed is supplied by the specifications and values that follow the *ADD command, and any applicable text or other material. Many of the data items that are stored in the database need not be explicitly coded in the *ADD Verb. Master default values are available from your Customer record, which is established and maintained by the System Operator.

As a convenience, the *ADD command may be abbreviated as *A.

***MOD**

*MOD affects changes to existing information in the VIS database.

It is important to note that *MOD is effectively a "delete-and-add" process. When a MODification is made to a Document:

1) existing information is merged with the requested changes,

2) the old database entry is physically removed, and

3) the changed entry is added as if it were a new Document.

The only exception to this process concerns Text Document MODification. Keyword specifications are NOT preserved and re-entered into the new entry if any Keyword changes are requested. in other words, any changes to the Keyword specifications replaces the existing specifications entirely. Note that if no KEYWORD= specification appears in a MODifying Verb, the original specifications will be deleted.

As a convenience, the *MOD command can be abbreviated as *M.

***KILL**

The *KILL command removes Documents from the VIS database.

*KILL is an unceremonious process, and should be used carefully, especially when dealing with billable-type Documents. Although each Document access is logged in the VIS response log file, the reference is to the Document's number and not its Title. Once a Document is KILLed and its number re-used by another Title, the original association is lost. For this reason it is recommended that billing information be extracted prior to KILLing active Documents.

When a document is KILLed, the document title, as well as some additional information from the **raw.** file, are written to the **/verb.bad** directory with the message ERROR 74 appended that indicates that the KILL was successful. This is merely a confirmation message to acknowledge the successful deletion.

As a convenience, the *KILL command can be abbreviated as *K.

***LIST**

The *LIST command produces a listing of the contents of a specified database entry.

The results of a *LIST request are written to a file in the VIS file directory **/vis.output**. This file is given the name **raw.** plus the first ten characters of the Document Title. You can either view the LISTing through your terminal or output it to the printer.

The output of a *LIST request is in the format of a Verb that defines the Document being LISTed. This output can be used to correct or reconstruct a faulty or lost Document.
For Text and Input Documents the body of the Document is included as part of the LISTing. Program and Graphic Document contents are written to separate files in the same directory (**/vis.output**) under the name consisting of the first 14 characters of the Document Title.

As a convenience, the *LIST command can be abbreviated as *L.

***REPORT**

The *REPORT command is used to request one of the database analysis utility reports. As a convenience, the *REPORT command can be abbreviated as *R. These reports are discussed in detail under the "Report Utilities" section.

Verb Specifications

The specifications and their values follow the command. They can be coded one after the other separated by spaces or tab characters, but it is recommended for clarity and ease of maintenance that you code them as individual lines.

The list of specifications varies according to the specific maintenance command and Document type. For each command/type combination there are certain specificaticns that are required, others are optional.

The Verb specifications and their values can be entered in any order. The only exception to this is the TYPE= specification which must follow the Edit/Maintain command as the first specification in the list.

Here are the specifications used in preparing Verbs:

TYPE= (required)

This specification identifies the type of Document to be added, changed, etc. TYPE= is ALWAYS specified and it MUST be the first specification following the command. Valid types are:

T	Text
P	Program
G	Graphic String
S	Synonym
M	User Master
D	User Detail
I	Input

Example: TYPE=T

ID= (required for User Master and User Detail verbs)

This is a four-position value that uniquely identifies a VIS User. Valid IDs include upper case alphabetic and/or numeric characters. It is required in Verbs dealing with User Master and User Detail maintenance. This value is prefixed with your VIS "ownership" value to form the User's Logon ID.

Example: ID=LANT

NAME= (required for all types except User Detail)

This specification means different things in different contexts. In a User Master Verb it refers to the name of the User, and it can be from 1 to 26 alphabetic or numeric characters enclosed in quotes. In a Text, Program, Graphic or Input Verb it represents the Document Title, and can be 1 to 25 characters enclosed in quotes. In a Synonym Verb it is the name of the Synonym, and is 1 to 15 characters **without quotes.** In all cases the NAME= specification is required in all Verbs.

Example: TYPE=T
NAME="TREES"

ORGANIZATION= (optional)

This specification appears only in a User Master Verb. It names the organization with which the User is associated. Its value is enclosed in quotes and can be up to 25 characters long.

Example: ORGANIZATION="ABC Corporation"
Default: none

ADDRESS= (optional)

User Master specification, 1 to 25 characters in quotes.

Example: ADDRESS="123 Ridgeway"
Default: none

CITY= (optional)

User Master specification, up to 20 characters enclosed in quotes.

Example: CITY="Anaheim"
Default: none

STATE= (optional)

User Master specification, 2 alphabetic characters.

Example: STATE=CA
Default: none

ZIP= (optional)

User Master specification, 5 to 9 position zip code.

Example: ZIP=92506
Default: none

AREA-CODE= (optional)

User Master specification, 3 position area code.

Example: AREA-CODE=714
Default: none

PHONE= (optional)

User Master specification, 7 position phone number without punctuation.

Example: PHONE=8531212
Default: none

PASSWORD= (required in User Detail type)

A User Detail specification only, this one to eight alphanumeric character value sets the password that the User is required to enter correctly at logon time. User Detail entries are uniquely identified by their ID= and PASSWORD= values so that a single User can have several different access levels from a single remote site depending on the password he specifies at Logon time.

Example: PASSWORD=ABCD
Default: none

TIME-OUT-SECONDS= (optional)

This specification is involved in User Detail maintenance. It specifies the time, in tenths of seconds, that VIS will wait for the User to respond before hanging up his line. More information on timeouts can be found in the section "Timeouts and Retries".

Example: TIME-OUT-SECONDS=300
Default: Customer Master specification

RETRIES= (optional)

A User Detail specification, this specification tells how many times VIS will re-prompt the User for a correct logon ID/password before hanging up his line. This value is also used as a retry limit during Menu response processing when no ELSE clause is specified (this is discussed earlier in the section on Text Programmed Presentation) as well as during program downloading in response to the "Load?" prompt. More information on retries, in relation to timeouts, can be found in the section "Timeouts and Retries".

Example: RETRIES=3
Default: Customer Master specification

PAUSE= (optional)

User Detail specification, tells VIS how long to wait (in tenths of seconds) between pages or screens of a Document. If the pause value is set to zero, (e.g., PAUSE=0), this will result in the maximum pause available, (about an hour), until the User terminates it by hitting any key. If a User sets their pause value to zero with '*' command (detailed in the User Guide), this sets the pause value to the value specified in the User Detail file.

Example: PAUSE=10
Default: Customer Master specification

SECURITY= (optional)

This specification sets the Security Code (Group and Level) for a Document or User. It is coded in the format GG,LL (two position Security Group and two position Security Level separated by a comma).

Example: SECURITY=15,00
Default: Customer Master specification

TERMINAL= (optional)

User Detail specification identifying the terminal type that the User will be working with. The value is three numeric characters. Each three-digit value points to a set of coded terminal device characteristics (graphics capability, screen capacity, etc) stored in a VIS file maintained by the System Operator. At Logon time VIS uses the codes to define how the terminal will be handled during the session.

Example: TERMINAL=100
Default: Customer Master specification

ONLY= (optional)

The characteristics mentioned above combine to define a particular terminal device's operating capabilities. A file containing sets of these definitions is maintained by the VIS System Operator. Changes and additions, if any, to this file are made as the need arises.

The entries in the terminal definition file are strings of codes identifying various characteristics and capabilities. The actual code values can conform to any desired convention (CompuServe, for instance) and/or be tailored to fit your specific needs.

When a User logs on, VIS attempts to identify his terminal by sending it a special character sequence and interpreting the response (if any). Typically, a terminal under control of Videotex software responds appropriately to this "inquiry"; terminals under "emulator" control generally do not. A valid response allows VIS to extract the "standard" set of characteristics from the terminal table.

If no response is made to the initial inquiry at Logon time, VIS uses the three-digit number you entered at User definition time in the TERMINAL= specification to extract a corresponding set of characteristics from the terminal table file. This terminal "profile" guides VIS processing during the User's session.

The basic format of a terminal table entry is:

id,code1,code2,...<CR>

The id field of the terminal table entry is a 3 to 5 character value that names either the standard CompuServe terminal code, or the three-digit TERMINAL= specification for the User.

These values include:

RS1	TRS-80 Model I, Level 1
RS2	TRS-80 Model I, Level 2
RS3	TRS-80 Model II, 12, and 16
RS4	TRS-80 Videotex Terminal
RS5	TRS-80 Color Computer
RS6	TRS-80 Model III
RSA	TRS-80 Model 4
RSD	TRS-80 Tandy 2000
000	TRS-80 DT-1 Data Terminal
100	TRS-80 Model 100
004	TRS-80 PT-210 Portable

You can add to either of the above sets as your circumstances warrant.

The following codes are used to identify some of the characteristics found among User terminal devices:

CC	Cursor Control
CW	Wide Character mode
CS	Standard Color control
CE	Extended Color control
G4	Semi-graphics 4
GH	High-resolution Graphics
GM	Medium-resolution Graphics
PB	CompuServe B Protocol
DT	Disk Transfer
MT	Memory Transfer ("arcade")
Wnn	Width 01-99 - characters per line
Hnn	Height 01-99 - lines per screen

You can add your own codes to the above list to identify any other characteristics you find useful in normal VIS processing. Contact the System Operator if the need to do so arises.

The ONLY= specification applies to Text, Program, Graphic and Input Documents. Its value is up to 16 characters in length and is enclosed in quotes. It is a string of codes representing terminal device capabilities. Access to the Document being defined is restricted to terminals whose characteristics (as set at logon time by VIS) include those named in this string. There is no default for this specification, and it is not required. It allows you to direct Documents, particularly Programs and Graphics, to certain terminals independently of the User's TERMINAL= setting.

Example: ONLY="RS4"
Default: none

SPENT-YTD= (optional)

User Detail specification of amount spent year-to-date. It is coded in dollars and cents format with an implied decimal point with a maximum of five digits allowed.

Example: SPENT-YTD=11025
Default: 0

SPENT-CURRENT= (optional)

Same as above but for the current billing period.

Example: SPENT-CURRENT=2675
Default: 0

KEYWORD= (optional)

1 to 15 characters that specifies the Keyword 1) associated with a particular Text Document or 2) to which a Synonym refers.

Example: KEYWORD=PINES
Default: none

PRICE= (optional)

This value, in dollars and cents with a decimal point, specifies how much it costs to access a Document. No Document can cost more than $99.99 to view.

Example: PRICE=0.25
Default: Customer Master specifications

Verb Terminators

A Verb terminator, like a command, must begin with an * (asterisk), or the TPP Character override value. This character signals the end of the specification list and/or the end of the Verb.

Valid terminators are:

***END**	end of Verb
***TEXT**	end of Verb specs - the body of this Text or Input Document follows immediately as part of this Verb and terminates with a *END

*END

The *END terminator marks the physical end of a Verb. All Verbs must conclude with a *END terminator, otherwise Edit will consider the Verb to be in error. The need for such a strict requirement becomes evident in cases where multiple Verbs are concatenated into a single file.

*TEXT

Text and Input Documents share the distinction of being the only Documents whose actual contents can be included as part of their Verbs. The *TEXT terminator provides this option.

In cases where text is included in the Verb itself, the specifications and their values are followed by the *TEXT terminator. The Document text follows the *TEXT, and the *END marks the Verb's end.

Sample Verbs

ADD or MOD a User Master

User information is stored in two different types of entries: User Master and User Detail. There must be a User Master entry for each of your Users. There must be at least one User Detail entry, although multiple entries are permitted.

The format and specifications involved in ADDing or MODifying User Master entries are as follows:

```
*ADD or *MOD
TYPE=M
ID=1234
NAME="John Q. User"
ORGANIZATION="Manny's Markets"
ADDRESS="4321 N 52nd St"
CITY="Running Springs"
STATE=MT
ZIP=80712
AREA-CODE=306
PHONE=2241919
*END
```

KILL or LIST a User Master

The Verb format for KILLing or LISTing a User Master entry is:

```
*KILL or *LIST
TYPE=M
ID=1234
NAME="John Q. User"
*END
```

The KILLed User Master entry is placed in the Customer's **verb.bad** directory along with an ERROR 74 as an audit trail. Note that a User Master cannot be deleted unless all its User Detail entries are KILLed first.

ADD or MOD a User Detail Entry

All VIS Users (as defined by User Master entries) must have at least one User Detail entry in the VIS database. Multiple Detail entries for a given User are allowed. User Detail entries are ADDed or MODified as follows:

```
*ADD or *MOD
TYPE=D
ID=1234
PASSWORD=XYZ
TIME-OUT-SECONDS=200
RETRIES=3
PAUSE=200
SECURITY=5,5
TERMINAL=100
SPENT-YTD=1050
SPENT-CURRENT=500
*END
```

KILL or LIST a User Detail Entry

The Verb format for these is:

```
*KILL or *LIST
TYPE=D
ID=1234
PASSWORD=XYZ
*END
```

The KILLed User Detail entry is placed in the Customer's **verb.bad** directory along with an ERROR 74 as an audit trail.

ADD or MOD a Text Document

The Verb format and specifications for ADDing or MODifying a Text Document are:

```
*ADD or *MOD
TYPE=T
NAME="LOCALNEWS"
KEYWORD=NEWS
    .
(as many as you want)
    .
SECURITY=0,0
ONLY="RS233,GM"
PRICE=0.50
*END
   or
*TEXT
Local hardware magnate John Q. User announced today....
..and his wife....for tax evasion.
*END
```

Note that in the case of Text Documents the body of the text can be included within the Verb file itself using the *TEXT...*END facility. If the contents are coded in a separate Document file, the last entry of that file must be *END.

ADD or MOD a Program or Graphic String

ADDing or MODifying a Program or Graphic String involves the same Verb format and specifications:

```
*ADD or *MOD
TYPE=P or TYPE=G
NAME="BLACKJACK"
SECURITY=0,0
ONLY="RS5,G4"
PRICE=5.00
*END
```

Note that Programs and Graphic Strings do not have Keywords and/or Synonyms. The contents of these Document types must be in separate Document files.

ADD or MOD an Input Document

ADDing and MODifying an Input Document is similar to ADDing and MODifying a Program or Graphic String:

```
*ADD or *MOD
TYPE=I
NAME="REQUEST"
SECURITY=0,0
*END
  or
*TEXT
request line
*END
```

Note that KEYWORD= and PRICE= specifications do not apply to Input Documents since they are invoked by Title and any applicable charges are associated with the Documents invoked by the request, not the request itself. The contents of Input Documents (i.e. the specific VIS request line) can be included in the body of the Verb itself. If the contents are coded in a separate Document file, the last entry of that file must be *END.

ADD a Synonym

ADDing a Synonym is quite simple:

```
*ADD
TYPE=S
NAME=STORY
KEYWORD=NEWS
*END
```

Quotes ("") are not used around the synonym as defined by NAME=. Note that Synonyms don't get MODified.

KILL or LIST Text, Program, Graphic, Synonym, or Input

These can be lumped together since their Verbs are very similar in format and specifications:

```
*KILL or *LIST
TYPE=T or TYPE=P or TYPE=G or TYPE=S or TYPE=I
NAME="LOCALNEWS" or NAME=STORY or NAME="whatever"
*END
```

The KILLed document is placed in the Customer's **verb.bad** directory along with an ERROR 74 as an audit trail. Note that quotes ("") are not used around a synonym when the document is KILLed or LISTed.

ADD an Initial Input Document

An initial Input document is associated uniquely with each customer. The uniqueness is derived from the NAME, which consists of a concatenation of a User's Logon ID (e.g., P1111), Password (e.g., ABCD), and the word "INITIAL".

```
*ADD
TYPE=I
NAME="P1111ABCDINITIAL"
*TEXT
request line
*END
```

ADD or MOD a Logon Document

A logon document is associated uniquely with each Customer. The uniqueness is derived from the NAME, which consists of the Customer"s five character **(/ocode)** and the word "LOGON", (e.g., "PLANTLOGON").

```
*ADD
TYPE=T
NAME="PLANTLOGON"
*TEXT
Welcome to the Agronimist's Information Service.
     .
     .
*END
```

ADD or MOD a Logoff Document

A logoff document is associated uniquely with each Customer. The uniqueness is derived from the NAME, which consists of the Customer"s five character **(/ocode)** and the word "LOGOFF", (e.g., "PLANTLOGOFF").

```
*ADD
TYPE=T
NAME="PLANTLOGOFF"
*TEXT
Thank you for the using the Agronimists Information
Service.
     .
     .
*END
```

Supplying Your Materials to VIS

"Local" File Transfers

The term "local" refers to any operating environment that does not use phone lines to interface with the VIS Host. Whether your service operates "on-site" (as part of a private service) or "off-site" (as part of a mass market service), this approach assumes that your materials (on Scripsit or TRS-DOS II diskettes) are "hand" delivered to the Host site for processing.

The extent of your actual involvement in the procedure about to be outlined can range anywhere from simply handing your Scripsit diskettes over to the System Operator (unless you are he) to doing the whole thing yourself.

STEP 1: Convert your Scripsit files (Verbs and Document files) to TRS-DOS ASCII files.

Scripsit provides a utility to convert its files to TRS-DOS ASCII files and vice versa. Refer to the Scripsit manual and the discussion of this facility. The process usually proceeds as follows:

From within the Scripsit document...

<CTL-U>
<C>

(S)cripsit -> ASCII or (A)SCII -> Scripsit (S or A) ?
S
TRSDOS file name ?
DOCNAME1 <ENTER>

STEP 2: Collect your converted Scripsit (now TRS-DOS ASCII) files onto a single diskette.

Scripsit puts the converted files on its own diskette. A good practice, though not a necessary one, is to collect these files onto separate diskettes, one for Verb files and one for Document files. Use the **COPY** command under TRS-DOS to do this collection. Though there are several formats associated with this command, you will probably use a series of **COPY** statements that look like this:

```
COPY RAW.VERB1:0 TO 1
COPY DOCNAME1:0 TO 1
```

STEP 3: Convert the TRS-DOS (Version 2) files to TRS-DOS II (Version 4) files.

If you do not have a TRS-DOS II system available to do this conversion, then your involvement in the "local" transfer process ends here. Your System Operator will have to assume the duty of completing the task.

The TRS-DOS II (Version 4) operating system provides a convenient command to accomplish the TRS-DOS conversion. Refer to the discussion of the **FCOPY** command in the TRS-DOS II documentation for explicit syntax requirements. In general, the command will look like this:

```
FCOPY 1 TO 0 (ALL) <ENTER>
```

This command will copy (and convert) all the TRS-DOS files (except system files) from the diskette in drive 1 to the one in drive 0. This format is viable only if you have, as recommended, segregated your Verb and Document files onto separate diskettes.

If you copy/convert one file at a time, use

```
FCOPY RAW.VERB1:1 TO 0
FCOPY DOCNAME1:1 TO 0
```

STEP 4: Transfer the TRS-DOS II files to the VIS Host file system.

You will have to be "logged in" to the TRS-XENIX system to perform this transfer. Every Customer is defined on the VIS Host and assigned a unique login ID. Using the system console, enter your login ID following the "login:" prompt. If the console is already logged in under another user ID, <CTL-D> will log it off, and the "login:" prompt will appear.

You use the **tx** command to transfer your files. You can transfer a whole diskette's-worth of files with one **tx** command, or you can transfer one file at a time. If your Verbs and Documents are on separate diskettes, and you put the Verb diskette in drive 0 and the Document diskette in drive 1, the process can be accomplished with two simple commands:

```
$ tx -x :0 /vis/(ocode)/verb <ENTER>
$ tx -x :1 /vis/(ocode)/document <ENTER>
```

The $ (dollar sign) is the TRS-XENIX user prompt. **tx** is the command. The **-x** tells **tx** to change carriage return characters to line-feeds (a necessary concession). **:0** and/or **:1** are disk drive designations. The notations **/vis/(ocode)/verb** and **/vis/(ocode)/document** are TRS-XENIX file directory designations; Verbs and Document files must be placed in their own directories. The **(ocode)** component of the directory specifications identifies your particular sub-directory within the VIS file system (**/vis**). This value is set by the System Operator and is supplied to you by him.

If your Verbs and Document files are on the same diskette, each file will have to be named explicitly in a **tx** command as shown in the following statements (follow each with **<ENTER>**):

```
$ tx -x :0 RAW.VERB1 RAW.VERB2 /vis/(ocode)/verb
$ tx -x :0 DOC1 DOC2 DOC3 /vis/(ocode)/document
```

You can specify as many file names as you like using the above format.

Note that **tx** converts the file names specified to lower case letters, so that **RAW.VERB1** on your diskette becomes **raw.verb1** in the TRS-XENIX file system, **DOC1** becomes **doc1**, and so on.

The examples above assume that you have observed the naming conventions for Edit/Maintain files, e.g. Verbs begin with the prefix **RAW.** and Document files contain the first 14 characters of the Document Title (NAME= specification in the Verb). If you have chosen not to observe these standards up to this stage, you will have to go through an additional series of commands to make sure that your files are properly named.

It should be apparent that certain commands (**FCOPY** and **tx**) exhibit simple formats when applied to segregated Verb and Document files. If you elect not to separate your Verb and Document files onto separate diskettes and **tx** your files to a single directory, you will have to move the various files to their appropriate directories.

In both the above circumstances you must function as a TRS-XENIX user to complete the file placement and/or renaming. Refer to the discussion applying to TRS-XENIX users and the commands they use for supplying their materials to VIS.

"Remote" File Transfers

As a "remote" Customer, i.e. one who interfaces with the VIS Host through the phone system, interactive communication and file transmit and receive procedures are accomplished using the TRS-DOS TERMINAL feature and its "RAM buffer" facility. It is recommended that you familiarize yourself with the documentation regarding these items since the discussion that follows is not an exhaustive discourse, but a procedural outline.

It is assumed that your equipment is installed and functioning, including your Model 12 (or other TRS-80 microcomputer), your high-speed (1200 baud) modem, your phone line, and all the necessary connecting cables. Installation and operation documentation should be referred to during the set-up and use of these pieces of hardware.

As a general sequence of events, your "remote" file transfer session will proceed as follows:

STEP 1: Convert your Scripsit files (Verbs and Document files) to TRS-DOS ASCII files.

Scripsit provides a utility to convert its files to TRS-DOS ASCII files and vice versa. Refer to the Scripsit manual and the discussion of this facility. The process usually proceeds as follows:

From within the Scripsit document...

<CTL-U>
<C>

(S)cripsit -> ASCII or (A)SCII -> Scripsit (S or A) ?
S
TRSDOS file name ?
DOCNAME1 <ENTER>

STEP 2: Establish a phone link with the VIS Host.

If your machine is off, power it up (and your modem, too), and "boot" into TRS-DOS (not Scripsit). If you are using Scripsit, press **<CTL-Q>** to exit the current document, if necessary, **<E>** to "end" Scripsit, and **<J> <ENTER>** to "jump" to TRS-DOS.

From the TRS-DOS READY prompt, type

TERMINAL <ENTER>

The TERMINAL Menu appears, and you are prompted with

--Enter Menu Selection ..

Select TERMINAL's TRS-DOS command interface specification by typing

S <ENTER>

When prompted for your command, type

SETCOM A=(1200,8,N,1) <ENTER>

You will then return to the TERMINAL Menu prompt. The only other option that should be set is the "ignore line-feed" option. TRS-DOS automatically generates line-feeds after carriage return characters sent to the display, so toggle this option ON to ignore the redundant line-feeds sent by the Host. To do this, type

L <ENTER>

The status of the option will be shown (it should be ON) and the Menu will prompt for another selection.

Dial the phone number provided by the System Operator. When the Host's modem sends you an audible tone ("carrier" signal), connect your modem to the line. The exact procedure for this varies for each type of modem, so consult your owner's documentation for details. Your modem must be in "originate" mode. When your modem responds with a second, somewhat garbled tone, hang up the phone receiver (or the equivalent) and go to your terminal.

If you have trouble making a connection, try the procedure again, making sure that all hardware is active and properly installed. If you have no luck at all, contact the System Operator for advice.

In preparation for communication with the VIS Host, put your terminal into "interactive" mode by selecting the following Menu option:

T <ENTER>

The terminal screen will clear and the cursor will assume its "home" position (upper left corner).

STEP 3: Login to TRS-XENIX.

Press **<ENTER>**. You should see

login:

on your screen. This is the TRS-XENIX login prompt. Respond by typing your TRS-XENIX user login ID (supplied by the System Operator), and press **<ENTER>**. If a password is required, enter it as well (the characters will not be "echoed" on your screen).

TRS-XENIX will continue prompting you until you login successfully. A successful login yields the standard TRS-XENIX greeting and the TRS-XENIX user prompt. When you see the following, your file transfers can begin:

$

STEP 4: Transmit your Verb and Document files.

Verb and Document files need not be segregated on separate diskettes since each file is transferred individually. It is necessary that they be in ASCII format.

A special format of the TRS-XENIX command **cat**, short for "concatenate", is used to transfer the input from your terminal into TRS-XENIX files:

$ cat >/directory/filename <ENTER>

where "$" is the TRS-XENIX prompt, **cat** is the command, and **>/directory/filename** indicates that input is to be directed to a file in the designated directory under the specified name.

The cursor will jump to the next line on your screen in expectation of input. Instead of typing the expected input on the terminal keyboard, however, you tell the TERMINAL program to transmit the input from its RAM buffer, which you load with the appropriate data. The following will transfer a Verb file from your diskette of converted ASCII files to the appropriate Edit/Maintain directory:

```
$ cat >/vis/(ocode)/verb/raw.verbl <ENTER>
<BREAK>          ...re-enters TERMINAL Menu mode
--Enter Menu Selection G <ENTER>
Enter Filespec (1-34)
RAW.VERB1 <ENTER>
```

TERMINAL loads the file into its RAM buffer and returns to its Menu prompt.

```
--Enter Menu Selection X <ENTER>
```

transmits the contents of the RAM buffer as if it were typed from the keyboard (except much faster), and returns to "interactive" mode when complete. Now press

<ENTER> and **<CTL-D>**

and you should return to the user prompt

```
$
```

You repeat this process for each Verb and Document file that you want to be processed during the next Edit/Maintain run.

STEP 5: Log off TRS-XENIX, disconnect from the Host, and exit TERMINAL.

Once all your Edit/Maintain materials have been transferred to their proper places with the appropriate names, you should log off the TRS-XENIX system. Following the user prompt, type

```
$ <CTL-D>
```

The TRS-XENIX login prompt should appear:

```
login:
```

Disconnect your line from the VIS Host by "hanging up" the phone.

Now exit the TERMINAL program as follows:

```
<BREAK>          ..re-enters TERMINAL Menu
--Enter Menu Selection Q <ENTER>
TRS-DOS READY
.....................................................
```

Some Useful TRS-XENIX Commands

When you "login" to TRS-XENIX, the system automatically positions you in your own file area, or "home" directory, and prompts you with $ (dollar sign). It is from this point that file preparation and manipulation begins.

Changing Your "Current" Directory

You can stay in your "home" directory or position yourself in one of the other available directories by using the **cd** command. Its format is:

cd /directory/ <ENTER>

where **/directory** is the general term that you replace with an appropriate directory name, like

/vis/(ocode)/verb.bad
/vis/(ocode)/document.bad
/vis/(ocode)/vis.output
/vis/(ocode)/report

Copying and Moving Files

When all your Verbs and Document files have been prepared, they must be supplied to Edit/Maintain before they can be processed. The system command **cp** is used to copy files from one directory to another, **mv** is used both to move files to different directories and to rename files within a directory, while **ln** is used to link another file name to an existing file. The command you use depends on whether you stay in your "home" directory or position yourself (using the **cd** command) in one of Edit/Maintain's directories to do your Verb and/or Document file preparation and manipulation.

The following cp commands show how to put your materials into their appropriate places while preserving a copy of each file in the directory from which the originals are being copied (Edit/Maintain purges its raw material files after it completes):

$ cp raw.anyverb /vis/(ocode)/verb <ENTER>
$ cp anydoc /vis/(ocode)/document <ENTER>

As an alternate to copying the file, the system command **ln** will create a synonym for the name of an existing file. Then, this linked name can be used for processing while preserving the original file and file name. The advantage of the **ln** command over the **cp** command is that the **cp** creates a new file with its own disk space requirements. If a Verb file has already been created in the proper directory, you can link another file name to it, preparatory to Edit/Maintain processing, with the following command:

$ ln anyverb raw.anyverb <ENTER>

If you do not want to "preserve" the original Verbs or Document files, **mv** simply puts files into other directories and does not copy their contents. If certain Verbs are not in the right directory, use this basic statement:

$ mv raw.verb1 raw.verb2 /vis/(ocode)/verb <ENTER>

Or if Document files need to be transferred to their own directory, use this one:

$ mv doc1 doc2 doc3 /vis/(ocode)/document <ENTER>

In each of these statements you can specify as many file names as you like.

The **mv** command is also used to rename files that do not yet conform to Edit/Maintain standards. Verb files are the most likely to be given non-standard names, so, assuming that you are in the **/vis/(ocode)/verb** directory, you would use the following statement to rename a Verb file:

$ mv myverb1 raw.myverb1 <ENTER>

Locating Your Files

The **ls** command can be used to see what files are present in any particular directory. Its format is

$ ls /directory <ENTER>

where **/directory** can be one of the following designations:

/vis/(ocode)/verb.bad
/vis/(ocode)/document.bad
/vis/(ocode)/vis.output
/vis/(ocode)/report

Whichever method you use to copy and/or move your files (**cp** or **mv**), you can verify the results by displaying the contents of the various directories by using the **ls** command:

```
$ ls /vis/(ocode)/verb <ENTER>
$ ls /vis/(ocode)/document <ENTER>
```

The diagnostic and report output materials are placed in specific directories during the Edit/Maintain (or Report) run. Display the appropriate directories by typing one of the following:

```
$ ls /vis/(ocode)/verb.bad
$ ls /vis/(ocode)/document.bad
$ ls /vis/(ocode)/vis.output
$ ls /vis/(ocode)/report
```

Once you know what files are involved, you can proceed with any necessary displaying and/or printing activities.

Displaying Files

Two commands are available for displaying TRS-XENIX files: **cat** and **pr**. Both produce results that can be viewed directly on your terminal screen or "re-directed" to the system line printer (discussed shortly) to obtain "hard copy" listings.

cat, short for "concatenate", is useful for displaying files that are not "printer ready", i.e. they do not contain print control characters (carriage returns, form feeds, etc). The basic format of the **cat** command is

```
$ cat filename <ENTER>
```

To display multiple files with one command, this format could be used:

```
$ cat file1 file2 file3 <ENTER>
```

Or to display all the files in a given directory, use this format:

```
$ cat /directory/* <ENTER>
```

where **/directory** can be any of your Edit/Maintain directory designations, and * indicates "all files" within that directory.

The **cat** command can be used to actually concatenate a number of files into a single file, which could then be formatted for printing or "downloaded" as a unit. The format of this **cat** statement might look like this:

$ cat /vis/(ocode)/verb.bad/* >bad.verbs <ENTER>

where **>bad.verbs** tells **cat** to output its results to the file **bad.verbs.**

pr, "print", produces printer-ready output from non-printer-ready input (like bad Verb and Document files) by including page numbers, headings, page breaks, date and time stamps, etc. To display the **bad.verbs** file just produced above in printer format, you would type

$ pr bad.verbs <ENTER>

This **cat/pr** combination could be used in a single statement using the TRS-XENIX "pipe" facility:

$ cat /vis/(ocode)/document.bad/* | pr <ENTER>

In this example, the results of the **cat** command are provided to **pr** as direct input using the | (vertical bar) symbol.

The variations and combinations of these (and other) TRS-XENIX commands are numerous. An examination of the TRS-XENIX documentation and a lot of "hands on" experience would be of great benefit at this stage.

For "remote" Customers who wish to "download" their materials for examination (or printing) at their own sites, use the TRS-DOS TERMINAL program's "RAM buffer" facility to capture the results of the various display commands shown (and implied) above.

Type the command line following the TRS-XENIX prompt. Before pressing **<ENTER>** to submit the request, follow this procedure:

<BREAK> re-enter TERMINAL Menu mode
--Enter Menu Selection **R <ENTER>**
RAM Buffer Now OPEN
Reset RAM Buffer (Y/N) **Y <ENTER>**
--Enter Menu Selection **T <ENTER>**

Your screen will clear. Press **<ENTER>**. You will see the material displayed on your screen as it is being written to the RAM Buffer. When the **$** re-appears

```
<BREAK>                to re-enter TERMINAL Menu
--Enter Menu Selection R <ENTER>
RAM Buffer Now CLOSED
--Enter Menu Selection C <ENTER>
Enter Filename (1-34)
FILENAME <ENTER>
```

The contents of the RAM Buffer will be copied to the specified file on your diskette. When the process completes

```
--Enter Menu Selection T <ENTER>
```

returns you to "interactive" mode. You can continue with other operations, or log off and disconnect from the Host as described in STEP 9 of "Remote File Transfers".

Printing Files

Both **cat** and **pr** produce results that can be directed to the system line printer to obtain "hard copy" listings of either unformatted or formatted materials respectively. Likewise, "printer ready" material, such as LIST or REPORT output files, can be routed to the line printer without additional manipulation. The **lpr** function under TRS-XENIX is used in all cases to "queue" items to be printed on the system line printer. The actual listings can be distributed to you later when they have been printed.

```
$ cat /vis/(ocode)/verb.bad/* | lpr <ENTER>
$ pr bad.verbs | lpr <ENTER>
```

These two statements show the use of **cat** and **pr** in conjunction with **lpr.** In each case, one involving unformatted material, the other formatted, the results are supplied to **lpr** by virtue of the | (or "pipe") symbol. As the line printer becomes available, the actual "hard copy" listings will be produced.

For LIST and REPORT output the **lpr** command is much simpler:

```
$ lpr /vis/(ocode)/vis.output/* <ENTER>
$ lpr /vis/(ocode)/report/* <ENTER>
```

Note again that * in the above statements indicates that all the files in the named directories are to be queued for printing.

File Clean-up

Once you have displayed all you need to display, and printed all you need to print, a little "housekeeping" is in order. To keep your file areas from becoming cluttered and unmanageable, it is recommended that you "clean up" un-needed files. The **rm** command is used for this purpose.

To "remove" a file from a specific directory, say a bad Verb from your bad Verb directory, use **rm** like this:

```
$ rm /vis/(ocode)/verb.bad/raw.verb1 <ENTER>
```

To delete all the files from each of your Edit/Maintain directories, use these formats:

```
$ rm /vis/(ocode)/verb.bad/* <ENTER>
$ rm /vis/(ocode)/document.bad/* <ENTER>
$ rm /vis/(ocode)/vis.output/* <ENTER>
$ rm /vis/(ocode)/report/* <ENTER>
```

Note that **/verb**, **/document**, and **/edit.output** are maintained by Edit/Maintain and do not require additional housekeeping.

Correcting and Re-submitting Files

Using the information provided in the diagnostic files produced during Edit/Maintain, you should be able to effect "repairs" to your faulty Verbs and Document files.

Chances are, the majority of your problems will be simple typographical mishaps or specification oversights, and these will probably occur during your learning period and virtually disappear as you become experienced and comfortable with the process.

You can return to the original Verb to make the appropriate corrections, or, if it is not available, you can copy or move (**cp** or **mv**) the particular **/verb.bad** or **document.bad** file to a work area and make the corrections or modifications there.

Once all necessary corrections have been made, the re-fortified files can be supplied to VIS, along with any new items, in the manner described in detail earlier.

The next Edit/Maintain run begins the cycle all over again.

What Can Go Wrong?

Both hardware and software are susceptible to occasional malfunction. Hardware failures should be mentioned for the effects they have on you and your Users. Software problems typically express themselves in the form of unexpected results encountered during normal use. Other situations, like customer billing, require the capability of extracting and analyzing response data stored in the VIS database. This section deals with problem contingencies, shutdown procedures, and diagnostic and information utilities.

It is important to note that your relationship with VIS is quite different from that of your Users. Your activities produce the contents of the service to which your Users subscribe. Your interface is much more intimately involved with the Host operating system, and because of this the problems that you experience during the course of your work with VIS should be brought to the attention of the VIS System Operator.

This discussion is more generally concerned with your Users and the problems and questions they direct your way.

Hardware Failures

Recall the physical components of VIS: the computer, multiplexer and on-line storage devices at the Host site, your terminal device (and modem, if you're remote), your Users' terminals and modems, and all the connecting cables and phone lines.

There is little to be mentioned with regard to the resolution of equipment problems at the sites remote from the VIS Host system. Owner manuals typically provide some material on problem determination and correction, but when these avenues dead-end, individual or group equipment maintenance agreements must guide the course of events at the particular remote site.

Problems with the phone system should be handled by the VIS System Operator in response to direct User contact, although you can assume the responsibility yourself. In any case, your Users should know who to contact in the event of telephone line problems.

Hardware failures at the VIS Host site obviously disrupt service throughout the network. You can be assured that affecting repairs at such critical times will be a top priority item! But even more unnerving than occasional interruptions of service, at least from a user standpoint, is the state of not knowing what's going on when they occur.

Shutdown

Shutdown refers to the process of disabling User access to VIS. There are two types of Shutdowns: Planned and Immediate, depending upon the urgency of the situation at the Host site.

Planned Shutdown

Planned Shutdown is possible when the urgency of the situation affords the time to execute it. Scheduled maintenance, minor software difficulties, and equipment problems that may potentially cause service interruption are examples of such situations. When a Planned Shutdown occurs:

1. No new incoming calls are answered, and
2. Users currently on the system are allowed to complete their outstanding requests, then after logging off the line is disconnected.

Typically, a Planned Shutdown will be initiated just before an Immediate Shutdown. Users whose requests do not complete during this period will be interrupted by Immediate Shutdown.

Immediate Shutdown

An Immediate Shutdown occurs when all VIS processing is halted, whether for reasons of hardware failure or software disorder, or the need to perform maintenance. In some cases, circumstances may be such that VIS is unable to execute anything at all. When an Immediate Shutdown occurs, all lines are disconnected.

Hardware failures and unscheduled system Shutdowns will be rare occurrences. Unexpected results that your Users encounter during their sessions with VIS will constitute the majority of your problem calls. Most of these will be simple misunderstandings of the use of VIS, particularly during the introductory sessions of the "novelty" period mentioned earlier. But some situations may require closer investigation.

Report Utilities

It is possible, however unlikely, that the connections between Titles, Keywords and Synonyms, or the links between chained Documents (as specified in the various TPP Code parameters) can become severed. A power-out during maintenance could cause such a situation. Your Users may encounter inconsistencies during their sessions with VIS that will prompt you to investigate and analyze parts or all of your Document structure.

The *LIST command of Edit/Maintain can provide information concerning Titles and their Keywords and Synonyms and the Keywords to which they refer. In addition to these aids, there are currently four utility reports that can assist you in tracing the interconnections within your Document structure:

```
SYNONYMS FOR KEYWORDS
TITLES FOR KEYWORDS
TITLES USED-BY
WHERE-USED
```

These requests are requested via the same procedures used for all other VIS maintenance: Verbs are prepared and supplied to the Edit program. Edit produces "object" material that is used as input to the Report program (not Maintain as in other maintenance functions). Output is in the form of "printer-ready" files written to your /report directory. You can either view the reports at your terminal (using the **cat** command) or output them to the printer (using the **lpr** function). These procedures are described in "Some Useful TRS-XENIX Commands".

SYNONYMS FOR KEYWORDS

This report shows all the Synonyms for a Keyword. The Verb to request it looks like this:

```
*REPORT
USED=SYNONYMS
KEYWORD=Keyword
*END
```

The report looks like this:

```
Customer Name                                    DATE:  5/24/83
CUSTOMER #00149                                  PAGE: 1

                SYNONYMS FOR KEYWORDS REPORT

        KEYWORD                    SYNONYM
        =======                    =======
        TREES                      FOREST

             1 synonyms found for keyword
```

TITLES FOR KEYWORDS

This report shows all the Titles associated with a particular Keyword. You request it like this:

```
*REPORT
USED=TITLES
KEYWORD=Keyword
*END
```

The format of the report is:

```
Customer Name                                    DATE:  5/24/83
CUSTOMER #00149                                  PAGE: 1

                TITLES FOR KEYWORDS REPORT

        KEYWORD                    TITLES
        =======                    ======
        TREES                      SPRUCE
                                   OAK
                                   DOGWOOD
                                   ELM
                                   HICKORY
                                   PINE
                                   MAGNOLIA

             7 titles found for keyword
```

TITLES USED-BY

This report shows for a particular Title all the Documents to which VIS may be directed while processing the MENU, WHEN, INCL, and LOAD TPP Codes. In effect, it maps the paths that a User may take from requesting the Document to a final end-of-Document. To request this report, specify

```
*REPORT
USED=BY
NAME="Title" or NAME=ALL
*END
```

The report format is

```
Customer Name                                    DATE:  5/24/83
CUSTOMER #00149                                  PAGE: 1

                        TITLES USED BY REPORT

 NAME                ESC  TITLE          TYPE        ERROR
 ====                ===  =====          ====        =====
M01                  MENU 0117ZX         TEXT
                     MENU 0113AW         TEXT
                     MENU 0115HT         TEXT
                     MENU 0121CA         TEXT
                     MENU INVRESP        TEXT

0115HT               CASE A0115HT        TEXT
                     CASE B0115HT        TEXT

1=LEAF NOT FND COMPONENT        4=DOC PTR NOT FND PARENT
2=DOC PTR NOT FND COMPONENT     5=FRAME NOT FND PARENT
3=FRAME NOT FND COMPONENT
```

WHERE-USED

This report is a simple "inversion" of the USED-BY report. It shows where (in what Documents) a particular Title is used as an argument of a MENU, WHEN, INCL, or LOAD TPP Code. To request the report, submit the following Verb:

```
*REPORT
USED=WHERE
NAME=ALL
*END
```

The format of the report is

Customer Name DATE: 5/24/83
CUSTOMER #00149 PAGE: 1

TITLES WHERE USED REPORT

NAME	ESC	TITLE	TYPE	ERROR
01172ZX	MENU	M01	TEXT	
0113AW	MENU	M01	TEXT	
0115HT	MENU	M01	TEXT	
0121CA	MENU	M01	TEXT	
INVRESP	MENU	M01	TEXT	
A0115HT	CASE	0115HT	TEXT	
B0115HT	CASE	0115HT	TEXT	

1=LEAF NOT FND COMPONENT
2=DOC PTR NOT FND COMPONENT
3=FRAME NOT FND COMPONENT
4=DOC PTR NOT FND PARENT
5=FRAME NOT FND PARENT

Appendix A: Error Messages

The following sections contain descriptions of the various error messages generated by Edit and Maintain. Included as well for reference, are vis_monitor User error messages that a User may encounter during a VIS session.

Edit Error Messages

The following are error messages generated when executing Edit. Generally, when Edit detects an error and writes the verb to the /verb.bad directory, the error message appears immediately after the line where the error occurred. The error message in the verb consists of a number that correlates to the numbers listed below. There is some overlap between the Maintain and Edit error messages.

31. The first character in a verb must be a TPP "escape" character.

All verbs start with a command and all commands start with an "escape" character, usually '*'. Check to make sure you are using a valid "escape" character for the command.

32. Invalid command - the valid commands are: ADD, MOD, KILL, LIST, or REPORT.

Check to make sure the command is specified correctly.

33. Document "TYPE" expected, but not found.

The "TYPE" option is required for every verb and must be the second line specified in the verb. Check to make sure this option is specified correctly.

34. Invalid option - "TYPE" option is incorrect.

Check to make sure the document "TYPE" specified is one of the valid types.

38. Invalid option - no such option available.

Check the documentation for the list of valid options for the specified document type.

39. Invalid option - option is duplicated.

An option in a verb has been specified twice. Check options for validity and remove one.

42. Invalid option - "SECURITY" option must be an integer between 0 and 15.

Check the syntax and values of the "SECURITY" option.

43. Invalid option - "SECURITY" option values must be separated by a comma.

Check the syntax and values of the "SECURITY" option.

46. Invalid option - "PRICE" option value must be in the range 0 - 99.99.

Check the syntax and values of the "PRICE" option.

49. Keyword missing - "TEXT" or "END" must follow options.

Check the verb syntax. After all options have been specified, the verb is either terminated with the keyword "END", or is followed by a document body, beginning the keyword "TEXT".

50. The document body was not located in the document directory.

Check for the presence of the document body in the /document directory. Also, Edit expects the document body file name to follow certain conventions.

53. Invalid command.

Check the syntax of the command. All "escape" commands must be followed by a space before their arguments.

55. Keyword missing - "END" must follow body of INPUT or VIS document.

Check to make sure the document body is terminated with the keyword "END".

59. Invalid data entry - this field may contain only numeric characters.

Check the indicated field for valid values.

60. Invalid data entry - this field may contain only alpha characters, numeric characters, or dots.

Check the indicated field for valid values.

61. Invalid data entry - this field may contain only alphanumeric characters.

Check the indicated field for valid values.

62. Invalid data entry - this field may contain only alpha characters.

Check the indicated field for valid values.

63. Syntax error - missing end quote or value too long.

The field must be terminated with an ending quote - if the value input exceeds the number of characters allocated for that field, the quote will not be found.

64. Invalid data entry - this field may contain only numeric characters and a decimal point.

Check the indicated field for valid values.

65. Invalid data entry - this field may contain only spaces or alpha characters.

Check the indicated field for valid values.

67. Keyword missing - unexpected end-of-file before "END" found.

All verbs and document bodies must be terminated with the keyword "END". Check the document body for this keyword.

69. Document body has extra characters after "END".

Edit expects all document bodies to be terminated by the keyword "END". Any characters after the "END" keyword in the file are considered erroneous. Check the document body for extra characters after "END".

70. Read was not successful.

In attempting to read from a file, a system error occurred. Try again, and if error persists, contact your system support representative.

71. Write was not successful.

In attempting to write to a file, a system error occurred. Try again, and if error persists, contact your system support representative.

75. Unsuccessful attempt to open a file.

In attempting to open a file for access, a system error occurred. Try again, and if error persists, contact your system support representative.

76. Unsuccessful attempt to close a file.

In attempting to close a file after access, a system error occurred. Try again, and if error persists, contact your system support representative.

77. No Information. This menu is incomplete.

The document file pointed to by the menu choice has not been processed yet by VIS.

200. Syntax error - "DATE" format must be either "mmddyy" or "mm/dd/yy".

Check the syntax for the "DATE" option.

201. TPP Code error - "INCL" arguments must begin with "(", (open parenthesis).

Check syntax of TPP code "INCL" in document body. The list of arguments following "INCL" must be enclosed in parentheses.

202. Invalid "REPORT" type requested - valid types are: NOT, WHERE, BY, SYNONYMS, or TITLES.

Check documentation for correct syntax of "REPORT" verbs.

203. Syntax error - missing closing parenthesis.

Check syntax of indicated field.

204. TPP Code error - illegal quotation marks.

Quotation marks inside the argument list of a TPP code are used only for document titles. Check the argument list for compliance.

205. TPP Code error - "REM" arguments must begin with "(", (open parenthesis).

Check syntax of TPP code "INCL" in document body. The list of arguments following "INCL" must be enclcsed in parentheses.

206. TPP Code error - "LOAD" arguments must begin with "(", (open parenthesis).

Check syntax of TPP code "INCL" in document body. The list of arguments following "INCL" must be enclosed in parentheses.

207. Keyword missing - "END" not found at end of verb.

All verbs and document bodies must be terminated with the keyword "END". Check the verb for this keyword.

208. TPP Code error - argument too long.

One of the TPP code arguments is too long for the field alloted to it. Check the syntax for the specific TPP code.

210. Syntax error - illegal character in "TITLE".

The "TITLE" contains a character that VIS does not allow. Check the documentation for the list of valid characters allowed by VIS.

211. TPP Code error - "MENU" arguments must begin with "(", (cpen parenthesis).

Check syntax of TPP code "MENU" in document body. The list of arguments following "MENU" must be enclosed in parentheses.

213. TPP Code error - "MENU" arguments must contain at least one value.

When presenting a "MENU", the argument list must contain at least one document "TITLE". Check the "MENU" argument list for compliance.

214. TPP Code error - "MENU" arguments contain illegal characters.

The "TITLE"s in the "MENU" arguments are subject to the same restrictions as other titles and can contain no special characters.

215. TPP Code error - "MENU" arguments are missing integer keys.

An integer must be associated with each "TITLE" in the list of "MENU" arguments, except in the case of the "ELSE" clause. Check the "MENU" arguments for compliance.

216. Access Denied.

An attempt was made to access a Customer Document other than a Customer's own. This is not allowed by VIS.

217. TPP Code error - TPP Code arguments must begin with "(", (open parenthesis).

Check syntax of the TPP code in the document body. The list of arguments following the TPP code must be enclosed in parentheses.

218. TPP Code error - "WHEN" arguments must contain at least one value.

Proper syntax for a "WHEN" argument list requires at least one document "TITLE". Check the "WHEN" argument list for compliance.

219. TPP Code error - missing leading quotation marks for title.

"TITLE"s inside argument lists for TPP codes require quotation marks. Check TPP code argument list for compliance.

221. TPP Code error - missing closing parenthesis.

Check syntax of TPP code inside document body. The arguments must be enclosed by parentheses.

222. Required option is missing for command/type.

Check documentation for requirements of the specific command or document "TYPE".

223. TPP Code error - "REM" argument too long.

One of the "REM" arguments is too long for the field alloted to it.

226. Unable to create file.

In attempting to create a file, a system error occurred. Try again, and if error persists, notify your system service representative.

241. TPP Code error - illegal argument for "LOAD".

An invalid argument was used in the argument of the TPP code "LOAD".

245. System Error during read.

A system error occurred in the middle of reading a file. Try again, and if the error persists, contact your system support representative.

249. Invalid option - "NAME" option for "REPORT" command.

The "NAME" option for the "REPORT" command must be either the word 'ALL' or a "TITLE".

250. Unable to access customer directory.

Check to make sure all directories are present.

Maintain Error Messages

The following are error messages generated when executing Maintain. Generally, when Maintain detects an error and writes the verb to the /verb.bad directory, the error message is written at the bottom of the file. This error message in the verb consists of a number that correlates to the numbers listed below. There is some overlap between the Maintain and Edit error messages.

70. Read was not successful.

In attempting to read from a file, a system error occurred. Try again, and if error persists, contact your system support representative.

71. Write was not successful.

In attempting to write to a file, a system error occurred. Try again, and if error persists, contact your system support representative.

72. This record already exists in the data base.

In attempting to ADD a document to the data base, Maintain discovered there was one already present with the same name. Possible solutions include deleting the old document or changing the name of the new one.

73. This record was not found in the data base.

In attempting to MOD, KILL, or LIST a document frcm the data base, the document was not located. Check to make sure you have the correct name for the document.

74. This record was successfully deleted.

This merely indicates that the document specified was successfully deleted from the data base.

75. Unsuccessful attempt to open a file.

In attempting to open a file for access, a system error occurred. Try again, and if error persists, contact your system support representative.

77. No Information. This menu is incomplete.

The document file pointed to by the menu choice has not been processed yet by VIS.

78. File was not located.

An unsuccessful attempt to locate a data base file occurred. Try again, and if error persists, contact your system support representative.

84. The option "TYPE" specified does not match that found in the data base.

In attempting to MOD, KILL, or LIST a document, a mis-match occurred between the "TYPE" specified in the verb and the "TYPE" specified when the document was added to the data base. Check to make sure you are specifying the correct "TYPE".

86. Invalid relational operator.

This is a system error. Try again, and if error persists, contact your system support representative.

87. Invalid logfile access.

This is a system error that occurs when an incorrect file is accessed. Try again, and if it persists, contact your system support representative.

91. There is insufficient buffer space available.

In attempting to allocate some buffer space in memory, there was an insufficient amount. Try again, and if error persists, contact your system support representative.

95. Keyword for synonym was not found.

In attempting to ADD a synonym to a keyword in the data base, the keyword was not located. Check to make sure you are using the correct keyword.

250. Unable to access customer directory.

Check to make sure all directories are present.

Vis_monitor User Error Messages

The following errors are those that are sent by vis_monitor to a User during a VIS session if something should go wrong. They are included here, as well as the User Guide, as a reference.

1. INVALID LOGON !!

Either the Logon ID or the Password (or both) were incorrect. Try again with the correct responses.

2. NUMBER OF LOGON TRIES EXCEEDED THE LIMIT !!

There is a limit on how many incorrect responses will be accepted when logging on before VIS severs the connection. Check to make sure you have the proper Logon ID and Password.

3. No information is available, please try another request.

For some reason, the information requested is not available. Possible reasons include: no information in the data base, invalid security codes, the document was designed incorrectly, or the document may be in the middle of an update.

4. Improper terminal type for data, please try another request.

The terminal being used can not correctly interpret the type of data stored in the requested document.

5. Command syntax is invalid, please try again.

Check the syntax for the specific command.

6. System error, please try again.

Try the request again; if the error persists, try another request. Contact your system service representative if the error continues.

7. System error, please try another request.

Try another request. If the error persists, contact your system service representative.

8. Menu response was invalid, please try again.

Check the menu for the valid responses and then try again.

9. Error in document, please try another request.

The information document requested contains an error. Inform your system service representative which document was erroneous.

10. All responses for menu prompt were invalid, please try again later.

There is a limit on how many incorrect responses the menu prompt will accept before returning to the "VIS:" prompt. Check the menu to make sure you are responding with a valid response.

11. Unable to "browse" further in this direction; please try another request.

The "browse" command (".+" or ".-") has reached an end of the data base and can not go further in the same direction. Reversing the browse direction, or repositioning in the data base with another request, should allow browsing to continue.

12. No information is currently available, please try again later.

The information document requested is currently being updated and is unavailable for viewing. Try requesting it again later when the update is completed.

13. Requested program is currently being loaded for another user. Please wait, or hit <BREAK> key to return to "VIS:" prompt.

Only one user at a time can have the same program loaded. If another user is loading the requested program, it may take a significant amount of time before the program becomes available.

14. System error in retrieving document.

This error means the data base has probably been corrupted and is inaccessible. Contact your system service representative as soon as possible.

15. All responses for "LOAD:" prompt were invalid; please try again later.

There is a limit on how many incorrect responses the "LOAD:" prompt will accept before returning to the "VIS:" prompt. Check to make sure you are responding properly to the "LOAD:" prompt.

16. Maximum of 99 documents may be requested at one time.

If a request would result in more than 99 documents being returned from the data base, only the first 99 will be available to be displayed.

NUMERICAL REFERENCE OF EDIT/MAINTAIN MESSAGES

Msg. No.			Msg. No.		
31.	Edit		86.		Maintain
32.	Edit		87.		Maintain
33.	Edit		91.		Maintain
34.	Edit		95.		Maintain
38.	Edit		200.	Edit	
39.	Edit		201.	Edit	
42.	Edit		202.	Edit	
43.	Edit		203.	Edit	
46.	Edit		204.	Edit	
49.	Edit		205.	Edit	
50.	Edit		206.	Edit	
53.	Edit		207.	Edit	
55.	Edit		208.	Edit	
59.	Edit		210.	Edit	
60.	Edit		211.	Edit	
61.	Edit		213.	Edit	
62.	Edit		214.	Edit	
63.	Edit		215.	Edit	
64.	Edit		216.	Edit	
65.	Edit		217.	Edit	
67.	Edit		218.	Edit	
69.	Edit		219.	Edit	
70.	Edit	Maintain	221.	Edit	
71.	Edit	Maintain	222.	Edit	
72.		Maintain	223.	Edit	
73		Maintain	226.	Edit	
74		Maintain	241.	Edit	
75.	Edit	Maintain	245.	Edit	
76.	Edit		249.	Edit	
77.	Edit	Maintain	250.	Edit	Maintain
78.		Maintain			
84.		Maintain			