The Inside View

The purpose of this document is to provide information on 'How to compile a working View system'. Details are given on 'where the files are kept', 'what is required to make them work', and 'how to make View compatible mail utilities'.

Files:

In order to compile View, you must be using Turbo Pascal version 6.0 from Borland. Other versions may work, but I only guarantee the latest version. You will also need some extra UNITs. Units are pascal libraries giving you extra commands. The units required are: novell.pas, textlink.pas and the Turbo Technojock Library. The TTT5 library can be found on BIZ2 in **\hamster\pascal\tttpas.arc**. The source for View can be found on BIZ2 in **\view\source*.***.

If you are starting from scratch, install Turbo Pascal 6.0 onto your hard drive. UNarc TTTPAS.ARC into a subdirectory. Follow the TTT doc file on how to compile the units. Copy novell.pas, and textlink.pas into your source directory. Compile them also. Once compiled, they can be used by any program. Each program is compiled separately. The only kink in the system is VIEW.PAS. The source for view contains code for both the network and ham versions. Near the top of VIEW.PAS you will see a {$DEFINE NOVELL}. If it is there, then a network compatible version is generated. If you remove it, or change it in any way, then a HAM (stand-alone) copy is generated. All other utilities only have 1 version and can be used by either version of View.

File Formats:

View does not deliver the mail. View creates messages to be sent, and reads messages already placed in your mailbox. The POSTMAN delivers the mail and notifies users of new mail. If you want to write mail utilities, then you must know how to create mailer files. Mailer files are used to move a message from sender to receiver. Postman does not know or care about which application generated the files. The rest of this document will describe what is required from a program to generate proper mailer files.

Mail format: View creates Internet compatible mail files. That means the header contains the neccessary info in the proper format to be excepted by any Internet mail system. Information on 'mail header formats' is available from most Network Information Centres including CCS at the UWO Natural Sciences Building.

Location: View uses the novell mail directories for the placement of both incoming and outgoing messages. For example: MBRAMWEL has SYS:MAIL/F003F

Sequence numbers: Every time a message a generated by View, a complete set of files is created for each recipient. For example, if you send a message to both Mark and Steve, two completely separate sets of files are created, 1 for Mark and 1 for Steve, even though the content is the same. This is because it is possible for one message to be delivered ok, whereas the second message might get 'bounced' back to the sender.

View uses 'sequence' numbers to maintain a sense of order with the files. Each 'set' has an unique sequence number. The 'last number used' is stored in SEQUENCE.SEQ. If this file does not exist, it is safe to assume that we can use the number 1 (unless it is locked). Both Postman and View locks files before openning them, and it unlocks the same files when done. To make things work, you should check for these file locks. To lock SEQUENCE.SEQ, create SEQUENCE.LCK in the same directory. It does not matter if there is any data in the file. If SEQUENCE.LCK already exists, then wait until it disappears before touching the sequence file. In my programs, I usually stop waiting after 10 seconds. If the program takes more than 10 seconds, then the machine it was running on has probally hung. After you have successfully created SEQUENCE.LCK; open SEQUENCE.SEQ; read the number stored it in; add one to the number; rewrite SEQUENCE.SEQ with the new number; erase SEQUENCE.LCK. We now have our own unique sequence number to use for the current message. You should make the

'get_sequence_number' routine as fast as possible so that other programs don't have to wait for you.

NOTE: Since the number is an integer, View can not handle very high numbers, therefore you may want to reset the number back down to 1 if it gets too high. I usually reset it after 32,000 messages.

Locking: Now that we have our sequence number, we must lock it. If we do not lock the number, Postman will try to process the files before we are done. Locking is accomplished by creating ####.LCK. For the rest of this document, I will use 1234 as our sequence number. Once we have created 1234.LCK, our programs can as much time as required to build the rest of the files. Postman will ignore all files associated with that number until the lockfile is erased.

TXT files: The whole message is contained in 1234.TXT. 1234 is the sequence number obtained in the previous step. Postman does not read the message file, instead it reads the .WRK file to determine the recipient. Your program should generate a proper message including all header info.

WRK files: This is a very important file. It tells the mailer who created the message, who should receive it, and where to send it next. 1234.WRK contains three lines. An example follows:

Example 1:
uwovax.uwo.ca
mark@novell.business.uwo.ca
alvin@uwovax.uwo.ca

Example 2:
ria.uwo.ca
mark@novell.business.uwo.ca
alvin@uwovax.uwo.ca

The 1st line contains the full hostname of where to send the message next. It is possible that the final destination is not reachable from your location. Therefore you may want the message to be 'punted' through a smart gateway(example 2).

The 2nd line is your full email address including the userid@hostname.

The 3rd line is the full email address for the recipient including the userid, hostname plus any other routing info (uucp uses the form of site1!site2! site3@hostname)

Wrapping Up: Now that we have created 1234.TXT and 1234.WRK, we can erase the 1234.LCK file. We are finished with that peice of mail. Postman will now try to deliver it to the recipient.

Summary: In conclusion, we have access the following files: SEQUENCE.SEQ, SEQUENCE.LCK, 1234.LCK, 1234.WRK, 1234.TXT. 1234.TXT and 1234.WRK contains a fully ready to go mail message.