

=====

GENERAL PLANNING CONSULTANT
TECHNICAL MEMORANDUM NO. 87.3.7
CLIST EXPANSION -- SUMMARY OF CHANGES
AND PROGRAMMER'S GUIDE

=====

Prepared for:
Southern California Rapid Transit District

Prepared by:
Schimpeler Corradino Associates

In Association With

The Cordoba Corporation
Myra L. Frank and Associates
Manuel Padron Associates
The Planning Group

MTA LIBRARY August 1987

S.C.R.T.D. LIBRARY

CLIST EXPANSION -- SUMMARY OF CHANGES AND PROGRAMMER'S GUIDE

The purpose of this Technical Memorandum is to summarize the changes made to the CLIST procedures on the SCRTD computer during FY87, and to provide a programmer's guide to the CLIST procedures and their supporting libraries.

1.0 INTRODUCTION TO THE CLIST PROCEDURES

CLIST procedures are programs written in the TSO command language, CLIST, that execute in the foreground on a mainframe terminal and enter into a real time dialogue with the user at the keyboard. CLIST procedures can request various inputs from the keyboard user and can perform any of the functions normally available in TSO, such as editing files, saving or deleting files, submitting batch jobs, etc. The primary advantage to using a CLIST procedure is that it allows a user to prepare and submit computer jobs without working directly with the Job Control Language (JCL) or the required input files of a particular program. Instead, a JCL file is prebuilt with dummy names for the data sets and other parameters that must be reset on each execution of the program. Input data files can also be prebuilt and dummy names used also for any user-specified values. In each case, the CLIST procedure is then written to ask the keyboard user to input the current names for each of the dummy-named parameters, and these are edited into the JCL and input files during the dialogue session. The resulting edits may be saved or not, depending on the structure of the CLIST procedure and the normal requirements of the batch jobs being run. At the conclusion of all editing, the CLIST procedure can optionally submit the job for running as a batch job.

There are two primary libraries of CLIST procedures on the SCRTD computer for use in transportation planning activities at the District. The first of these is designed to execute in the foreground a number of useful routines, such as printing a file either on the line printers in the District's Headquarters building or on a laser printer in the same building, providing status checks of the user's batch jobs, etc. The second library is accessed from this first one and contains CLIST procedures that edit and submit UTPS jobs for travel forecasts and simulations.

2.0 PROGRAMMER'S GUIDE

2.1 RELEVANT FILES AND LOCATIONS

The two primary files of CLIST procedures are:

MRP.TSO.CLIST VB format, 255 record length, location PLAN03 contains 36 members currently, RACF protected to permit changes to be made only by Keith Killough or Peter Stopher.

MRP.DRIVER.CLIST VB format, 255 record length, location PLAN02, contains 25 members currently, RACF protected as MRP.TSO.CLIST.

The members in MRP.TSO.CLIST access a variety of different JCL and procedural files that are located in various libraries and sequential data sets. Some of these use system procedures, while others use custom-written procedures. To locate the files accessed, the programmer is advised to go to the end of the CLIST file and page up until finding the file name that is edited or submitted by the CLIST procedure. Generally, this is to be found within one or two screens of the end of the CLIST file. Appendix A contains a list of members of MRP.TSO.CLIST, a brief description of what each member does, and identification of other files accessed, edited, or submitted by the member. Most of the members of MRP.TSO.CLIST have been developed and installed by Planning Department staff, although the core of the procedures was originally developed and installed by Schimpeler Corradino Associates under the Transportation Planning and Modeling Services Contract. Currently, the General Planning Consultant and Planning Department staff share the responsibility for maintaining these procedures and improving them.

The members in MRP.DRIVER.CLIST access JCL files that are stored in MRP.DRIVER.CNTL, usually under an identical member name. The details of the MRP.DRIVER.CNTL file are:

MRP.DRIVER.CNTL FB format, 80 record length, unnumbered files, location PLAN02, contains 51 members currently, RACF protected as MRP.TSO.CLIST.

In turn, most of the members of MRP.DRIVER.CNTL are JCL substitution in JCL setups for the UTPS programs, and the JCL file in which the substitutions are made is SYS1.URD79.EGOPROC. Appendix B contains a list of the members of MRP.DRIVER.CLIST and a short description of what each one does and the JCL file that is edited by it.

2.2 OPERATING CONVENTIONS FOR THE CLIST PROCEDURES

These CLIST files currently operate in two alternative ways. Predominantly, they are set up to ask the user to enter from the keyboard certain selected pieces of information required to set up and run a batch job. The user enters a response to each question, and the CLIST makes any appropriate checks to ensure that the input information is correct, such as checking the file catalogs to make sure that input files already exist and that output files do not exist. In many instances, the user is asked to confirm that the entry made is correct, by entering a yes or no response after the CLIST shows on the terminal the information it just received from the keyboard. Where there are limitations on the length of a string that the user can input, this length is checked and the user is asked to reenter the information if it lies outside the prescribed limits. If a requested input file is not present, the procedure will not permit the user to proceed. If an output file specified already exists, the user will usually be given the option to replace this file, or to use a different name and continue.

The second mode of operation, which is planned to be used more widely in the future, is a menu-driven version of the CLIST procedures. In these, the CLIST will display a screen of options, with associated numbers/letters and keywords. The user is asked to enter the number/letter or keyword of the desired option and the procedure continues with that option. Inputs, such as data set names, are still handled through keyboard entry, checking, and confirmation.

All of the CLIST procedures can be exited by using the "PA1" key, which returns control to the calling procedure or the TSO screen, if the procedure was initiated from there. In addition, some of the CLIST procedures will accept "Q" or "QUIT" when typed in response to a question in the procedure and will exit upon receiving that response. The CLIST procedures in MRP.DRIVER.CLIST have two additional features of importance. First, most of the jobs use tapes for output, for which a Job Memo is required by the system operator, prior to initiating execution of the job. The CLISTs are designed to edit a Job Memo data set and print two copies of it at the operators line printer. Second, there is a flow chart data set that constructs a schematic of the inputs, outputs, and program runs required in a full network simulation. The CLISTs construct a data base of information for inclusion in the flow chart for each named simulation. When a print of the flow chart is requested from the CLIST procedures, the appropriate data are entered into the schematic to provide a flow chart that shows the job numbers and dates of execution of each program in the UTPS stream, enters the computer account ID that was used to run each job, and shows the tape numbers of output data sets and the data set names of all input and output data sets for that simulation. The flow chart can be printed at any time in the program-running sequence and can be used to show how far a particular simulation has proceeded. It is automatically updated by each

successive program run, provided that the run is made from the CLIST drivers.

3.0 SUMMARY OF CHANGES MADE IN FY 1987

3.1 CHANGES TO EXISTING CLIST PROCEDURES

3.1.1 Tape Numbers

During FY 1986, the SCRTD computer system changed to a tape management system that permitted cataloging of all tape files by volume number, with retention of the tape catalogs on disk. This change required a major change in the CLIST procedures for UTPS, for which previously it had been necessary to obtain tape numbers prior to a job that would output tape files. Under the old system, the user would request from the tape librarian the assignment of specific tapes to be used for output and would enter these numbers in a Job Memo delivered to the system operator before the job would run. The CLISTs were set to utilize this procedure by having the user enter the tape numbers prior to submission of the job, thereby including these numbers on a Job Memo generated by the CLIST and storing them in the flowchart record for the UTPS simulation. Under the new system, tapes are no longer assigned in advance. When a job requires one or more tapes for output, the operator takes the next available tape from the library of unused tapes and the tape-cataloging system records the number of the tape with the data set and other labeling information at the time that the job is run successfully.

As a result of this change, a temporary fix was made during FY 1986 to the CLISTs to remove the requests for the tape numbers in the procedures. This meant, however, that the tape numbers were no longer available to be recorded in the flow chart. During FY 1987, all of the CLIST procedures that generate tape jobs were changed. The request for a tape number was reinstated in the procedures that used as input the tape(s) created in a prior step or that first used a tape provided from an external source. The request was preceded by a LISTC ENTRIES (dsn) VOLUME command that searches the catalog for the data set name and displays limited information about the data set, including the tape number. The result of the LISTC is displayed on the screen and the user is asked to key enter it, so that the CLIST procedure can read it into a variable name. In this way, the tape number records were reintroduced in the tracking of the jobs, and the flow charts again include the tape numbers. Currently, however, tape numbers are only recorded when the job that uses a tape as input is run. If the entire UTPS job stream is run, this is no problem, because the last job (ELD3R) uses the tapes created by the preceding job and does not create any new tape outputs. However, if the run is ended with mode-of-arrival, the mode-of-arrival output tape will not be entered into the flow chart data base.

3.1.2 Catalogues

The second change made during FY 1987 was to several of the CLISTs in MRP.DRIVER.CLIST that automatically produced catalogues of members of a partitioned data set, or catalogues of families of sequential data sets. When the CLISTs were written originally, these catalogues were generally quite short and filled one or perhaps two screens. After several years in operation, they had expanded to large numbers of screens full of data set names. The user had no option to avoid the listing and no way to stop it in the middle. The change made was to make all such catalogues optional at the user's request, and to restrict some of them by specifying a larger part of the data set name. This reduced considerably the amount of time required to execute some of the CLISTs and also reduced user frustration, which is exponentially related to nonproductive time at the terminal!

3.1.3 New PO Driver

The third change was made to the PO member of MRP.TSO.CLIST. This member allows the user to print off any data set or a member of a partitioned data set on the Planning Department's remote printer. The procedure had encountered some problems that resulted in an inability to produce more than one copy, even though the procedure was supposed to be able to generate multiple copies, and an inability to route the printout elsewhere than "Remote 1" (the Planning Department's remote printer) even though the procedure offered the option to send the printout elsewhere.

While fixing these problems, the driver was substantially rewritten to reduce the number of responses required from the user in 90 percent of cases. In the earlier design, the driver requested input of the LRECL of the data set to be printed, the number of copies, the printer location, an indication as to whether the data set contained carriage control characters, etc. In almost all cases, the responses to these questions are the same. Yet, the user was required to enter a number or a yes/no, and a confirmation yes/no for every one of these parameters on each submission. In addition, if more than one data set was to be printed, the user was required to go back and restart PO from the TSO screen.

The changes made were to place the parameters into a menu screen, with the defaults set as the most common responses. The user was asked to enter "N" or "NO" if no changes were required to the defaults, or a letter code indicating the parameter to be changed. The result of this change was to allow the user to obtain a printout of a conventional data set (1 copy, LRECL=80, destination=RMT1, no carriage control characters, etc.) with four keystrokes instead of a minimum of 15 keystrokes. The CLIST also executes in about half the time if no changes are required. Two further changes were made. First, a question is now asked at the end of the processing of the printout request as to whether the user wishes to print another data set. If

the answer is yes, then the CLIST continues in operation. If no, it returns control to the TSO screen. Second, if the user responds yes, then the data set name just printed is redisplayed. The user may now edit the displayed name for the next name, if he or she chooses, or may enter a complete new name. The ability to edit the last name makes it much less time-consuming to print off several members of a partitioned data set, as well as allowing families of similarly-named sequential data sets to be printed with much less effort.

3.1.4 Changes to ELD3 Set Ups

As a result of changes made in FY86 and early in FY87, including production of a new version of URAP (see Technical Memorandum 87.3.2- - Bus Statistics/Route Analysis Program Documentation -- URAP 2), the ELD3 driver had become completely inoperable and its JCL was no longer correct. The task was undertaken to return the ELD3 driver and its JCL to operable status. This involved some fairly extensive JCL modifications to conform to the new URAP program, and included creating a new URAP load module in MRP.DRIVER.LMOD. Having completed the JCL modifications, the CLIST program was modified to produce the correct edits of the JCL. With the task in the GPC Work Program to produce a mainframe version of the cost model in FY87 (a task that was subsequently dropped), reference to the old bus cost model, developed under the Transportation Planning and Modeling Services contract, was deleted temporarily.

3.1.5 Changes to MDIST Procedure

During FY86, a new MDIST procedure was constructed that computed distances between zone centroids and the nearest station using the coordinates and calculating a rectangular or straight line distance. This procedure was developed to be a substitute for the previous MDIST procedure that used the highway network to obtain the distances. The coordinate-based procedure had never been set up to run in the CLIST drivers and had been operated subsequently from outside the drivers. In FY87, a new driver was written in CLIST and a new JCL member created to use the revised MDIST procedure, which was also compiled into a load module in MRP.DRIVER.LMOD. The manual procedure had been known as XYMINDIST, but was used to replace the old MDIST procedure in the drivers.

3.1.6 Modifications to UFARE

As a first attempt at a procedure desired for most of the UTPS-based CLIST procedures, the UFARE CLIST member was modified extensively to allow the user to set UPARM parameters from the keyboard as part of the current submission. This involved the addition of a considerable amount of code in the CLIST procedure to edit each of the UPARMS. In this version, the default values of the UPARMS are displayed first and the user is asked if he or she wishes to change any of them. If not, processing continues to job submission. However, if the user wishes to change any of the UPARMS, they are displayed one at a time for

editing and the user is asked if he or she wishes to change each one. It is expected that future modifications will either permit selection of UPARMS for editing from a menu screen or that a full-screen editing capability may be added as an alternative.

3.1.7 Addition of the UEVAL Driver

A set of programs called UEVAL was developed and installed on the District's computers early in FY87. These procedures perform an evaluation of a simulation procedure and produce a number of important statistics to describe a simulation result and compare between two simulations (see Technical Memorandum 86.3.1 -- Documentation of the Measures of Effectiveness Evaluation Programs - UEVAL). Three steps were required to develop the UEVAL driver procedure. First, the programs were compiled as load modules in the partitioned data set MRP.DRIVER.LMOD. Second, a new JCL stream was created for MRP.DRIVER.CNTL for editing by the new CLIST procedure, and including appropriate variable names in the JCL. Third, a new CLIST procedure was written, conforming to the CLIST procedures in MRP.DRIVER.CLIST, and with addition to the main UDRIVE menu of a call to the UEVAL procedures. The CLIST was created as a single member, named MRP.DRIVER.CLIST(UEVAL). The JCL exists in six files, namely:

```
MRP.DRIVER.CNTL(EVAL1)
MRP.DRIVER.CNTL(EVAL2N)
MRP.DRIVER.CNTL(EVAL2W)
MRP.DRIVER.CNTL(EVAL3)
MRP.DRIVER.CNTL(EVAL4N)
MRP.DRIVER.CNTL(EVAL4W)
```

The load modules are in an identical set of members of MRP.DRIVER.LMOD. Operation of the new UEVAL program also required a modification to the UPRAS program, the load module for which is stored in MRP.DRIVER.LMOD(UPRAS2).

3.1.8 Addition of the FAREMTRX Driver

A new fare matrix program, FAREMTRX, was developed and installed on the SCRTD computer at the end of FY86 (see Technical Memorandum 87.3.1 -- Documentation of the Fare Matrix Calculation Program - FAREMTRX). This program is intended to replace the existing FARE80 and FARE00 members of UFARE as the standard fare program, although calibration to the validation year must yet be performed before this program becomes the standard part of the UTPS simulations. During FY87, similar steps were undertaken with this program to the UEVAL programs. A load module of the FORTRAN program was built in MRP.DRIVER.LMOD(FAREMTRX) and a new JCL stream was created in MRP.DRIVER.CNTL(FAREMTRX). A CLIST procedure was then developed to be included in the UTPS sequence that would edit the JCL in the FAREMTRX member of MRP.DRIVER.CNTL. The new driver is stored in MRP.DRIVER.CLIST(FAREMTRX).

3.1.9 Addition of the BUSES Driver

During FY87, a FORTRAN program was developed to create the data base for downloading to the microcomputer bus cost model. This program has been added to the MRP.DRIVER.CLIST procedures through the creation of a load module called MRP.DRIVER.LMOD(BUSES) and a JCL stream in MRP.DRIVER.CNTL(BUSES). The CLIST procedure edits the JCL and submits the program in the load module, according to the directions of the user. The CLIST is currently executed by the user entering the command in TSO "exec 'mrp.driver.clist(buses)'." It is planned that the procedure be accessible subsequently from the mrp commands by typing in the word "buses" after using "mrp" at the commencement of the terminal session. However, this change will not be executed until FY88.

3.1.10 Prototype of a Menu-Driven CLIST Procedure

The final activity undertaken during FY87 was the initial development of a prototypical CLIST driver that would be much more user-friendly than the existing ones. The additional friendliness would be provided by using a series of nested full-screen menus, from which the user is asked to choose options. Only those options chosen by the user are then implemented in the procedure. For all other conditions, defaults values are supplied by the procedure. The initial development of this prototypical procedure was undertaken in a copy of the EPNET CLIST, but is not yet fully operational. However, it was used as the model for the development of the UEVAL and FAREMTRX drivers, and will be used as the model for all future additions to the drivers and modifications to existing drivers.

3.2 DIRECTIONS FOR FUTURE CHANGES

Based on the changes made during FY87, it is expected that all of the drivers will be changed, little by little, into menu-driven procedures that also allow the user to edit any relevant UPARMS. The emphasis in revisions to the CLIST procedures will be on greater ease of use, omission of unnecessary user response, and greater speed of operation wherever possible. In addition, modifications are expected to be made to support documentation of alternative simulations through the FLOWCHRT procedure and through a memo describing each simulation. Changes will also be made in UDRIVE itself to keep it up to the standard of the added programs and to incorporate calls to additional procedures.

APPENDIX A
DETAILS OF MRP.TSO.CLIST

MEMBER NAME	DESCRIPTION	FILE SUBMITTED
DISKMAP	Prints a map of the contents of the planning disks	MRP.DRIVER.CNTL(DISKMAP)
INET	Delete transit network (TNET) data set 1-5	MRP.TNET.&YEARALT.&PERIOD.&CNT..DATA
IGLTEST	Used to test graphics	MRP.USKETCH.LOADLIB(PRWTEST)
JOBMEMO	Build a new jobmemo data set for job submission	MRP.DRIVER.CNTL(JOBMEMO)
KJ	Kill (delete) class=x printout from a job	SYSTEM UTILITY
LDLIB	Load a new/revised member into shared PDS library	SYS1.LINKLIB(IEBGENER)
MOVE	Move a sequential data set to a new allocation	SYS1.LINKLIB(IEBGENER)
MRP	Make Metro Rail Planning commands/help available to user	MRP.TSO.CLIST, SYS1.CMDPROC, SYS3.RTD.CLIB, MRP.HELP.DATA, SYS1.HELP
OJ	Display class=x job printout, retaining the printout	SYSTEM UTILITY
PC	List MRP subcatalog groups	LISTC
PJ	Transfer printout from class=x to class=0 for printing	SYSTEM UTILITY
PD	Print a data set (version 2)	MRP.DRIVER.CNTL(PRINT)
PDLASER	Print a data set using 4th Floor laser printer	MRPAKK.POLASER.LOADR(LISTC), MRPAKK.POLASER.LOADR(LISTW)
POLASERS	Print a data set using 5th Floor laser printer	MRPAKK.POLASER.LOADR(LISTC), MRPAKK.POLASER.LOADR(LISTW)
POLASERB	Printed output will contain up to 132 characters	MRPAKK.POLASER.LOADR(LISTC), MRPAKK.POLASER.LOADR(LISTW)
PODLD	Submits IEFTPCH to print a data set (version 1-not used)	MRP.DRIVER.CNTL(PRINT)
PREFIX	Rename data sets by changing the data set name prefix	SYSTEM UTILITY (RENAME)
PRM	Reset terminal session to normal logon parameters	SYS1.CMDPROC, SYS1.HELP
PT	Submits on IEFTPCH to print a data set	MRP.FORMAT.CNTL(PRINT)
RBLS	The SCRTO Ridership by Line Segment software	STG.MRP.CLIST, MRPAKK.RBLS.PLTRLIB
REBLK	Reblock a sequential, fixed block (FB) data set	SYS1.LINKLIB(IEBGENER)
ISC	List user subcatalogs	SYSTEM UTILITY (LISTC)
SJ	List status of u-drive production jobs for a flow chart	SYSTEM UTILITY (ST)
ISORT	Sort a data set (accessed via %sort) to PLAN02	SYS1.LINKLIB(ICEMAN)
TRNPLOT	Route Plot	MRP.USKETCH.LOADLIB(TRNPLOT), MRP.USKETCH.LOADLIB(DELSEG)
UORIVE	The SCRTO UTPS Transportation Planning Models	SYS1.LINKLIB(IEBGENER)
UORIVE2	Developmental UORIVE procedure (not operational)	SYS1.LINKLIB(IEBGENER)
UPLDT	Produces a UNET plot at the planning graphics terminal	MRP.URD79.PROGLIB(UTPSPRW)
UPLDT1	Allocate the CLIST libraries Allocate the panel libraries Allocate the message libraries	MRPKLK.PANELS, MRPKLK.SKTLIB, MRP.TSO.CLIST(UPLDT2)
UPLDT2	Used to build simulation network plots (under development)	MRP.UTPSLOT.LOADLIB(TUPLDT6C)
UPLDT3	Drives plot card: Multiple panels of data can be processed by this conversation in order that multiple plot cards may be created for a given INET run. Validation can be found in both the panel (UPLDT2) and the CLIST (UPLDT3).	NONE
USKETCH	The SCRTO multi-purpose mapping software	MRP.USKETCH.CNTL(DISKTAPE) MRP.USKETCH.CNTL(ONLINE)

APPENDIX B
DETAILS OF MRP.DRIVER.CLIST

MEMBER NAME	DESCRIPTION	FILE MANIPULATED
ACCOUNT	Programming aid for adding the request for an account number.	None
BUSES	Runs the FORTRAN program to create the data base for the GPC Bus Cost Model	MRP.DRIVER.CNTL(BUSES)
ELD3R	Submits ULOAD/URAP/UCOST models from the UTPS Model Package.	MRP.DRIVER.CNTL(ELD3C) MRP.DRIVER.CNTL(ELD3R) MRP.DRIVER.CNTL(ELD4C) MRP.DRIVER.CNTL(ELD4R)
EPNET	Submits the UTPS program UNET with the network files in a partitioned data set.	MRP.DRIVER.CNTL(EPNET)
ETNET	Submits the UTPS program UNET with the network files in sequential data sets.	MRP.DRIVER.CNTL(ETNET)
ETSKM	Submits the UTPS program UPATH four times and UPSUM once.	MRP.DRIVER.CNTL(ETSKM)
FAREMTRX	Runs the FORTRAN program to create peak and off-peak fares from the paths.	MRP.DRIVER.CNTL (FRMTRXMD), (FRMTRXWK), (FRMTRXKR), (FRMTRXPR)
FLWCHRT	Edits the current flow-chart and submits a PO job to print two copies at the RMT1 printer.	MRP.FLWCHRT.CLIST(&FCN) MRP.DRIVER.CNTL(PRINT)
JSTAT	Determine the status of jobs being run through the drivers under the specified flowchart.	System utility
KMCHWORK	A variant of MCHWORK.	MRP.DRIVER.CNTL(MCHWORK)
LISTC	Programmer's aid for performing a LISTC.	None
MCHNWRK	Submits the UMODEL setup for the HBO, OO, and OW models of SCAG/BAA	MRP.DRIVER.CNTL(MCHNWRK)

MEMBER NAME	DESCRIPTION	FILE MANIPULATED
MCHWORK	Submits the UMODEL setup for the SCAG/CSI HEW mode-choice model.	MRP.DRIVER.CNTL(MCHWORK)
MDIST	Runs the FORTRAN program to calculate minimum zone-to-station distances (Version 2.1).	MRP.DRIVER.CNTL(MDIST)
MDIST2	Version 2.0 of MDIST	MRP.DRIVER.CNTL(MDIST)
MOARRVL	Submits the UMODEL setup for the modified BAA mode-of-arrival model.	MRP.DRIVER.CNTL(MOARRVL)
NAMES	Contains a list of person names, office and home telephone numbers for use by Jobmemo for MRP ids.	None
NEWTAPE	Programmer's aid for setting up new tapes for input in CLISTs.	None
PARAM	Programmer's aid for CLIST to edit user parameters in UIPS setups.	None
UDRIVE	Duplicate of the UDRIVE member in MRP.TSO.CLIST	Not executed
UEVAL	Determine the status of jobs being run through the drivers under the specified flowchart.	MRP.DRIVER.CNTL(EVAL1), (EVAL2N), (EVAL2W), (EVAL3), (EVAL4N), (EVAL4W)
UFARE	Submits the UMODEL program to compute fare matrices using modules FARE80 or FARE00, as specified.	MRP.DRIVER.CNTL(UFARE)
UPARM	Programmer's aid for writing CLIST to set user parameters.	None

MEMBER NAME	DESCRIPTION	FILE MANIPULATED
USTOS	Submits the UTFS procedure USTOS (as amended by the GPC for more stations).	MRP.DRIVER.CNTL(USTOS)
VERSION	Programmer's aid for writing CLIST to show the version number of current CLIST programs.	None