UMS
Programmer’s Manual
Release 6.0
November 2002
## Table of Contents

### CHAPTER 1. UMS OVERVIEW
- PROSCRIPTIONS ................................................................. 1
- RELATED MANUALS ............................................................... 1
- OVERVIEW ........................................................................... 2

### CHAPTER 2. CONTROL-DISPATCH
- SYSTEM OVERVIEW ........................................................... 9
- General Overview .............................................................. 11
- Main Control ........................................................................ 11
- Host Interface ...................................................................... 12
- Output Services ................................................................... 12
- Mapping/Demapping Services ............................................. 12
- Secondary Session Services ................................................ 13
- Control/Dispatch Conventions ............................................. 13
- General Structure .............................................................. 18
- Guest Side Conventions ...................................................... 19
- Host Side Conventions ....................................................... 20
- Inter-Process Communication ............................................ 21
- Program Function-Key Standards ...................................... 23
- Guest-Host Control Blocks ............................................... 25
- Guest Common Area Structure .......................................... 27
- UMS Software-Interface Hooks ......................................... 30
- Host Common Area Structure ............................................. 32

### CHAPTER 3. UMS Naming and Version Conventions

### CHAPTER 4. UMS LIBRARIES AND MODULE PROMOTION
- Proposed Module Movement Changes ......................... 44

### CHAPTER 5. UMS PROGRAM CONTROL TABLES
- Program Control Table Entry Guest (PCTEG) .................... 53
  - Source Code Example - PCTEG ......................................... 54
  - The PCTEG In Detail ......................................................... 56
  - SYSPARM Options .......................................................... 58
  - Assembled Code Example ................................................ 61
  - Hexidecimal Dump ........................................................... 68
- Program Control Table Entry Host (PCTEH) ...................... 71
  - Source Code Example - PCTEH ......................................... 75
  - Assembled Code Example - PCTEH ................................. 77
  - Hexidecimal Dump of PCTEH Example .......................... 84
- Internal Function Codes .................................................... 87
  - Internal Function Codes List ............................................ 88

### CHAPTER 6. LXTABLE PROCESSING
- Feature Summary .............................................................. 91
- Detail Description ............................................................ 96
  - The LXTABLE Macro ...................................................... 96
  - The UMSLXTBL TYPE=START ...................................... 97
  - Sample expansion, TYPE=START ..................................... 98
  - The UMSLXTBL TYPE=DETAIL ...................................... 99
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Checking</td>
<td>104</td>
</tr>
<tr>
<td>Edit Checking</td>
<td>106</td>
</tr>
<tr>
<td>Edit Type Table</td>
<td>107</td>
</tr>
<tr>
<td>User Exit - LXTABLE</td>
<td>108</td>
</tr>
<tr>
<td>User Exit Example</td>
<td>111</td>
</tr>
<tr>
<td>Internal Field Values - UMSLXMAC</td>
<td>113</td>
</tr>
<tr>
<td>Sample Map Source Fragment</td>
<td>114</td>
</tr>
<tr>
<td>Sample Assembler Map Dsect</td>
<td>115</td>
</tr>
<tr>
<td>LXTABLE Assembled Example</td>
<td>117</td>
</tr>
<tr>
<td>Hexidecimal Dump of LXTABLE Example</td>
<td>137</td>
</tr>
<tr>
<td>CHAPTER 7. SYSTEM UTILITY PROGRAMS</td>
<td></td>
</tr>
<tr>
<td>Resident Utilities</td>
<td>139</td>
</tr>
<tr>
<td>Date Conversion Routine</td>
<td>140</td>
</tr>
<tr>
<td>Example of a COBOL Invocation</td>
<td>140</td>
</tr>
<tr>
<td>Data-Name Address Routine</td>
<td>141</td>
</tr>
<tr>
<td>Miscellaneous Edit Services</td>
<td>142</td>
</tr>
<tr>
<td>In-Core Online Sort</td>
<td>143</td>
</tr>
<tr>
<td>Non-Resident Utilities</td>
<td>144</td>
</tr>
<tr>
<td>Violation Table Processing</td>
<td>144</td>
</tr>
<tr>
<td>CHAPTER 8. CICS TABLES</td>
<td></td>
</tr>
<tr>
<td>UMS Setup Options</td>
<td>149</td>
</tr>
<tr>
<td>CHAPTER 9. UMS SCREEN MAPPING PROCEDURES</td>
<td></td>
</tr>
<tr>
<td>CHAPTER 10. UMS ONLINE ERROR MESSAGES</td>
<td></td>
</tr>
<tr>
<td>UGZ0004P - The Message Module</td>
<td>173</td>
</tr>
<tr>
<td>Assembled Example of a Message Module</td>
<td>175</td>
</tr>
<tr>
<td>Hexidecimal dump of Message Module Example</td>
<td>180</td>
</tr>
<tr>
<td>CHAPTER 11. SPECIAL SYSTEM FUNCTIONS</td>
<td></td>
</tr>
<tr>
<td>Limited Secondary Session</td>
<td>181</td>
</tr>
<tr>
<td>UMS Screen Hop Facility</td>
<td>182</td>
</tr>
<tr>
<td>ALARS-UMS Bridge</td>
<td>184</td>
</tr>
<tr>
<td>GHOST Program - ESI/NDR/CDLIS</td>
<td>186</td>
</tr>
<tr>
<td>CHAPTER 12. UMS BATCH</td>
<td></td>
</tr>
<tr>
<td>Tape Processing</td>
<td>191</td>
</tr>
<tr>
<td>CHAPTER 13. RECORD SURROGATES</td>
<td></td>
</tr>
<tr>
<td>APPENDIX A: NON-UMS LXTABLE EDIT EXAMPLE</td>
<td></td>
</tr>
<tr>
<td>APPENDIX B. GUEST PROCESS CONTROL TABLE</td>
<td></td>
</tr>
<tr>
<td>APPENDIX C. HOST PROCESS CONTROL TABLE</td>
<td></td>
</tr>
<tr>
<td>APPENDIX D. CICS MISCELLANEOUS</td>
<td></td>
</tr>
<tr>
<td>CICS Abend Codes List</td>
<td>261</td>
</tr>
<tr>
<td>CICS Queue Names</td>
<td>262</td>
</tr>
<tr>
<td>APPENDIX E. UMS SYSTEM PROGRAMS</td>
<td></td>
</tr>
<tr>
<td>GUEST-SIDE PROGRAMS</td>
<td>265</td>
</tr>
<tr>
<td>HOST-SIDE PROGRAMS</td>
<td>266</td>
</tr>
</tbody>
</table>
# Table of Contents

## Appendix F. Guest Common Area
- COBOL Example ................................................................. 267
- ASSEMBLER Example ....................................................... 272

## Appendix G. Function Descriptions
- Function Titles List .......................................................... 277

## Appendix H. CICS IDMS Network Relationships
- NETWORK CONNECT OVERVIEW ........................................ 379
  - PRODUCTION ENVIRONMENT ........................................... 379
  - TEST ENVIRONMENT ..................................................... 380
  - TRAINING ENVIRONMENT .............................................. 382
  - MRO ENVIRONMENT .................................................... 383
  - THE TMON V8.0 ENVIRONMENT ...................................... 384
- CICS - IDMS REGION RELATIONS ....................................... 385
  - PRODUCTION ENVIRONMENT ......................................... 385
  - TEST AND TRAINING ENVIRONMENT ............................... 386
1

UMS Overview

Proscriptions

This UMS PROGRAMMER’S MANUAL is provided to assist those users who have a definite requirement to modify or extend the UMS application software. There are five important principles to guide such modifications and/or extensions:

1. While the RMV provides problem determination and correction for the UMS Guest Software, it cannot provide such support to user-modified or extended Guest Programs.

2. The UMS System Control and Support Programs named UGZxxxxP may not be modified. The only exception is for those modules involved in site definition whose modifications are specified in the Install Manual.

3. All fields passed to the HOST for whatever purpose must successfully pass their LXTABLE edit where such an edit is provided. The list on page 107 lists the edits that must be employed. The section beginning on page 91 discusses the LXTABLE in detail. The Appendix beginning on page 203 provides discussion and examples of the LXTABLE use in programs outside of the UMS Control Structure.

4. Prior to making contact with the HOST, the UGTL transaction must have executed successfully in the current execution of the CICS region concerned.

5. Contact with the HOST will be effected only by means of the UMS System Interface programs supplied by the RMV. Please refer to #2 above.

In addition, please read the convention section beginning on page 13 and the interface section beginning on page 21.

Related Manuals

UMS Installation Manual: This book discusses the installation of the stated release. These instructions preview the UMS system from a “high level” viewpoint.
UMS Technical Manual: This manual discusses in detail each of the UMS transactions, processing, programs, screens, records, and error messages. This is the best source, outside of the programs themselves, of understanding the operation of the UMS functions. Appendix G draws a synopsis for each function from this manual.

UMS User Manual: This is a “How-To” for using the system. The audience for this manual is the End User.

UMS Customization Manual: This manual is a guide for modifying the UMS system distributed to the user community with the other manuals. It is a subset of this book.

Overview

The Uninsured Motorist System (UMS) was developed to allow insurance companies to notify the RMV when a policy on a vehicle had been cancelled, creating a possible uninsured motorist. The system operates in both the batch and on-line mode. Over time, enhancements have been made to the system to cover far more than insurance policy changes. It has intimate connections with ALARS (Automated Licensing and Renewal System) such that former distinctions are now blurred. To the same extent, the responsibilities for these systems have changed as well.

The UMS system is built on a client-server basis. The RMV computer is the server, or Host, for data on registrations, licensing, and related issues. Many of the clients or guests are insurance companies or “outside” agencies with their own data processing facilities and needs. UMS provides them with a complete interface and set of application programs to access the RMV data. They are free to make modifications to the UMS Guest programs. They are, however, required to use the UMS Structural and LXTABLE edit facilities so they can maintain a maximum posture of release independence and conform to data edit requirements. The system design places much of the workload on their machine. The UMS design makes simple requests of the Host database system so that the database run-units are short with as few I/Os as possible to give the very best available response for all users.

A client who either modifies the released software or creates its own assumes full responsibility for conforming to RMV data Standards.

A dozen CICS and IDMS regions provide the various classes of service required. The diagram on page 5 illustrates the relationships. In all cases, CICS is the teleprocessing and application front-end for the IDMS CVs. The database services are filtered through the Logical Record Facility (LRF) of IDMS. The General User CICS region is the LU6.2 contact point in the Host for “guest” users who have connect privileges during normal business hours.
Three general types of Guest-Host interface are in service. These services are illustrated in the following pages. In Type 1, the UMS Guest software resides in the Guest’s mainframe and uses LU6.2 services to reach the RMV computer (see page 6). The RMV host software does some validations, security checking, and the IDMS database I/O. The Type 2 interface is used by several official agencies (see page 7). In this, the Guest software runs in the RMV CPU and the LU6.2 facility is replaced by an interface program. The Type 3 interface is similar to Type 1 but either the machine is not an IBM or the user (guest) has heavily modified the “guest” software (see page 8). This is a case of a “black box” communicating via LU6.2. We have no particular responsibility for problems with this type of “guest” software.

For a more detailed description of functions or for error message information, consult the UMS Technical Manual for this release level.

The overall design objectives are to use single function, independent modules and segregate the front-end (CICS or TP) and back-end (IDMS or database) processing. This has been achieved through several subordinate objectives or techniques. The first UMS design objective is to place as much of the application work as possible in the Guest software. The second objective is to drive the Guest and Host software systems with an application control structure. This structure, called Control/Dispatch (CD), supports the third objective, which is that development and maintenance time for applications be minimized. CICS provides all the needed control facilities, but they must be hard-coded into the programs. The programs become larger in size and complexity, and changes to the processing structure cause changes to a number of modules. It is also difficult to run new and old versions of programs in parallel.

The UMS CD system is table driven so that control flow changes are only applied in one table. Common services, such as screen mapping, data movement, field editing, and error message handling are all directed by the CD system and occur in system modules outside of the application program. The chapters on Control Dispatch, UMS Program Control Tables, and LXTABLE Processing describe these control mechanisms in detail. Many utility routines are provided so that the application program can concentrate on the specific business function and just call for basic “services” such as date routines. Components are intentionally isolated so that debugging and error resolution is simplified and new functions can be added without changing or disturbing the existing ones.

The fourth objective is function-reuse. This applies particularly to the Host side functions that provide the database services. If a new requirement needs half of the data provided by Host-function “X” and additional data not provided anywhere else, a new Host function will be written to get the “additional data.” In this way, if there is a problem with the function “X” data, there is still only one place to look for the problem. There would be many possibilities if each new data requirement was met with a new function providing only a specific group of fields. Such a proliferation would be expensive in terms of development and a nearly insupportable nightmare in terms of maintenance and problem resolution.
The fifth objective is to make the interface to the DBMS as narrow and well defined as possible. This ensures that the impact of changing the DBMS would be minimal to the application systems. As a corollary, all programs must use the application “common type services,” which operate through specific interfaces. This aids in development and, in case of error or needed change, there is only one place to apply the correction and then all programs will be using the new code. This principle is extended further in cases such as license plate number edits where the edit rules set is elaborate and subject to sudden change. The edit code is in a separate routine accessible to all applications but it is driven by a set of rules that is downloaded from the Host at Guest start-up time. In this way, the rules could be changed daily with no visibility to the users at all. This design eliminates the need to manually distribute zaps or complete new copies of the rules tables and ensures that all users are totally and painlessly up to date.
TYPE 2 GUEST/HOST INTERFACE

User Agency Terminal

CICS

LX TABLE

IDMS

IDMS CV

GUEST OUTPUT

GUEST PROCESSES

HOST PROGRAMS

GUEST INPUT

HOST MAINFRAME (RMV)

- Signon-On
- Input Screen Editing
- Request Data-Base Info.
- Output Host Response

- Check Security
- Additional Editing
- Data-Base Retrieval
- Respond To Guest

SINGLE CPU CONFIGURATION
2  Control - Dispatch

System Overview

The UMS Guest System requires initialization once the CICS region’s startup is complete. The UGTL transaction invokes UGZ0015P to perform the various operations. This process is discussed in more detail in a subsequent chapter. If this process is not completed, UMS will abend with a code of “UGTL” or “UGTH.”

The transaction “UG03” is the normal entry to the UMS Guest Processing. It invokes UGZ0001P, the Guest Side Control Program. This routine controls the processing thread for the transaction. Application programs that are invoked usually return to this routine unless they abend. The control program examines the control structures and passes the thread to the next program or function for the task. The focus of control is the system portion of the common area and the PCTEG entry in UGZ0002P for the transaction in flight. The PCTEG is discussed in detail later. It contains a list of programs to be invoked in turn to do the work of the transaction. If the automatic screen mapping, demapping, and editing feature is to be used, the name of the (LX) control table is given. If a special clean-up program is needed, its name is given. Flags show the ability to do updates, scrolling, and re-scrolling as well as the ability to communicate with the Host System. Some additional features are discussed in the detailed treatment. The system may also be invoked by external programs that start the transaction UG05 (that points to UGZ0008P). UGZ0008P can be called directly by link or xctl.

At transaction initialization time, the data from the corresponding PCTEG entry is placed into the system portion of the Guest Common Area. The structure of the Guest (and Host) Common Areas are briefly covered in the following section of this chapter. The control program and the application both refer to the Guest Common as the transaction progresses. The application moves data elements only so a change of a program name is transparent and will only require one change in the PCTEG entry for that transaction. The order of the program names in the PCTEG is function dependent and the application can vary the name moved to the 'next program' field depending on run-time conditions.

The first named program in the PCTEG list is the first one called by the control program once mapping is complete. Any program can only be first for one transaction (and its test version, 1st character = ‘T’). When a screen of data is to be processed, the designated LX map table is loaded and the LX map program completes its work before that first application program is invoked. If the map module detects errors it can cycle through error-message issue and re-try the mapping until the input is error free. The data from the screen is placed into the dsect in the common area before the application gains control.
The program can edit the data further and issue error messages of its own to the screen which will continue this cycle to correctness.

A number of service routines are available to do common functions such as upper-case conversion and date conversions. These are described in Appendix G - Function Descriptions. Calls to the Host side for database services can be made for retrieval and/or update. Control can be returned to the caller or another program it designates. Each time the thread returns control to CICS (except for logoff), the control program sets the next transid to UG04. This id shows a continuing transaction to insure initialization that does not negate work already done. If access to the Host System is indicated a Host Interface Area block of storage is acquired and initialized by the Control Program. When the work is complete and data is written to the screen, the Control Program invokes the LX-Table feature to move the data to the screen. As is described in the LX-Table processing chapter, exit routines may be summoned during screen processing (in or out) to perform special editing. These routines can generate error messages.

Error messages are designated by a numeric value in Common Storage. This value is used in conjunction with UGZ0004P, the Error Message Dictionary, to retrieve the actual text and place it in the output map. When the thread comes back to the Control Program as completed, the designated clean-up program, normally UGZ0005P, issues the write to the terminal and makes any required final “adjustments.”

Each UMS screen normally has an LX table, guest input program, one or more host programs, and a guest output program. When the user enters data onto the screen, the LX table processor uses the LX table for that screen and edits the data. Next, the guest input program receives control to format the guest-to-host blocks (refer to the Guest-to-Host Blocks section) and performs any additional editing. The guest input program passes control to the host side by using an internal function code to indicate which host programs to invoke (refer to the Internal Function Codes section). The host programs perform database access and formats the needed data into the host-to-guest blocks. The guest output program receives the host-to-guest blocks, uses this data to make decisions about screen highlighting, and performs other miscellaneous tasks. The last programs to receive control are the LX table processor and the Clean up Program, UGZ0005P. These programs use the screen's LX table to map the data from the General Storage Area (GSA) to the screens, and sends the screen back to the user.
General Overview of UMS-Guest Control/ Dispatch

Control/Dispatch (often abbreviated CD) is the name given to the various service level functions provided for the UMS guest application programs. There are two purposes to CD. The first is to provide a level of standardization to functions common to multiple application areas. The second is to perform functions deemed overly complex for the typical application module.

General areas of functionality:

A. Screen/CRT/CICS - Map management
B. Memory management
C. Function validation
D. Host interface
E. Table services

Each of these areas is a component of CD, and may exist as a unique module, or as a component of a multi-purpose module(s).

Main Control

All UMS transactions are “wired” to the same PCT entry. UMS requires at least 2 transactions to run. One (typically UG03) is referred to as the initialization transaction. Another (typically UG04) is referred to as the default run transaction. When the main control program gets control, it determines if this is an initialization call by checking the transaction name against the initialization transaction name. If the initialization transaction is found, the common area is cleared, a flag indicating initialization is set, the current map-name is set to the logon map, and control is transferred to output services.

If no function is currently selected, output services are invoked to inform the user.

Note that when the user comes back through, the logon process is treated in a manner essentially the same as any other function, except that the user is required to complete logon before a function change is allowed.

UMS supports three mechanisms of saving COMMAREA (Application-high core, VSAM and CICS-high core). The main control module insures that the current COMMAREA image is in an area located below the line.

CD determines if the user has changed the function-code. If so, the new function code is validated. If it is valid and all required modules are present, the internal data is changed to cause the selected function to be dispatched. If an error is detected, output services are invoked to send the message to the user.

If no function is currently selected, output services are invoked to inform the user.
If a function key has been pressed, the meaning of the key is checked for validity in the current environment. If it is invalid, output services are invoked to send the error message to the user. Otherwise, the appropriate service is invoked.

The only remaining action is to dispatch the application. Most applications use table driven mapping/demapping (LXTBL) services, and a few do not. Those that do not use LXTBL services are transferred directly. Those that use LXTBL, require that services be dispatched before the application.

### Host Interface

When an application determines that it must obtain data from the host, it builds the application portion of a host parameter block and then transfers to the host interface module defined for the guest. The interface module checks for the host being active, completes the control portion of the host parameter block and transmits the data to the host. When the response is received, the host interface module transfers back to the requesting application. Note that if the host is not active, or if a fatal error is detected on the host side, the host interface program directly invokes output services to post an error message.

### Output Services

Output services has four activities to perform:

1. Format the common map header (date, time, etc)
2. Lookup any message code in the message dictionary and place the text in the map
3. Setup for saving the COMMAREA according to the site option for COMMAREA location
4. Determine the next transaction code (specified for the executing function, or the default run transaction)

Once these activities are complete, the map is sent, and the program returns.

### Mapping/Demapping Services

Most applications use these services for processing maps. These services provide for table driven transfer and editing of fields between the COMMAREA and the map buffer. Some fairly sophisticated cross-field edits and host table edits are available. If errors are detected, they result in the direct invocation of output services. Any application which uses these services for demapping also uses them for mapping. Transferring these services occurs before the application and before output services.
Secondary Session Services

This mechanism allows the user to temporarily leave the current function, perform another function, and return to the original. CD treats the secondary session as a toggled entity. If the second session is not active, the request must be made to activate; if it is active, the request must be made to terminate. Activation consists of saving the COMMAREA and current screen (via 3270-READBUF) in a temporary storage queue and going through normal dispatch. Termination consists of restoring the COMMAREA screen and going through normal output services.

Control/ Dispatch Conventions

This is a general set of guidelines, or rules, which must be followed in UMS applications in order to maintain the proper interface with the control/dispatch mechanism.

1. Module names are managed through the PCTE (G & H) entries in both the Guest and Host systems. All reference to specific modules in the applications is via cells in the common area loaded by control-dispatch. No application is to have any reference to specific module names.

2. Guest function names are managed through the PCTE entries. No reference to specific function names (except for internal-only names) is to be present in any set of application modules.

3. Host function (block-type) codes follow the same conventions as guest function names with respect to applications running on the host side. On the guest side, applications calling host functions must be cognizant of the appropriate host name, and seed it into the host interface area.

4. In cases where a guest function is responsible for the output side of a particular duplicate resolution, the parallel host function must be included in the appropriate guest PCTE entry. With this exception, there is no parallelism between host and guest PCTE entries.

5. All maps must be generated using the standard UMS map header macro. This results in all the UMS maps being identical in structure throughout the message area. The structure forces one map per mapset. Further, all transmission and receipt of terminal data is handled by the control-dispatch mechanism. This means that control-dispatch will be responsible for providing an application with an input-map when the application is invoked, and sending an output map when the application exits to the control functions. Control dispatch also provides an area in which the map variables may be placed by the application in concert with cells for map-name, cursor position, message-code, message-text and length of variable-data. This information allows proper map return.
Additionally, the LXTBL option provides a mechanism by which the control dispatch mechanism relates map-data, map-name and common-area fields automatically without specific application intervention. This feature provides for raw field editing, support of basic field types including character, numeric, date, expiration-date, internal zoned-decimal, internal packed-decimal, internal binary, and internal compressed-date. Further, operands are being included on the definition macros to provide for a wide range of field content editing and cross-field editing.

6. Control-dispatch is responsible for memory management. Application programs may not contain GETMAIN/FREEMAIN (CICS or MVS) requests.

On the guest side, Control dispatch saves and restores the common area across each pseudo-conversational interaction. Hooks exist on the guest side for the acquisition of additional work area for specific functions which are not retained across pseudo-conversational interactions. Such acquisition will be implemented when a definitive need arises.

7. Guest application programs for UMS are named in the PCTEG entries, and are expected to be concise units aimed at specific business functions. Broad based functionality is construed as being of a utility nature (such as the LXTBL mechanism, date-conversion, address-editing, common field-editing, etc) and being a function-dispatch service. Any services of this nature not currently meeting these general needs will be provided as requested.

The general idea is that the application set defined for a given business function will consist of 1 or more modules. The modules will be viewed as being between the processing of terminal-input and a host dialogue, followed by 1 (or possibly more) modules viewed as being between the receipt of a host response and terminal-output. In practice, it is intended to be possible (when testing modifications to a guest function) to run both old and new functions in the same region by having one PCTE pointing at the old, and another at the new. This clearly demands a narrow focus of application, with modules serving one business function only.

The functional-independence of the module structure is required not only to maintain these abilities, but also to insure that there is some reasonable potential for guest users to migrate the functionality of this code to their existing systems.

8. The same general set of criteria applies to the host. The host is viewed by the guest as a data-server which operates on a quick in and out basis with the guest. The host treats each guest interaction as a unique interaction, with no knowledge of any previous interaction. The host common-area is not retained across interactions with the guest.
There is no restriction against the guest going to the host more than once for a given transaction. The intent is to push all work except raw data-service onto the guest. Thus, all possible work is done at the guest side, and is not to be repeated at the host side.

9. The UMS control/dispatch contains mechanisms for interface with foreign (non-UMS) applications, such as ALAR, SYSM, etc as well as any others that may be running on the guest.

The specification of a business function as being external (foreign) occurs in the guest PCTE entry, where the XFER= option must be specified with a value of YES or DATA. Control may be passed to an external program or to another Task. The LXTABLE discussion covers this feature in detail.

10. UMS control-dispatch is able to accept control from foreign applications in the same fashion as it can initiate them. The following are available to foreign applications:

A. START the external initiation transaction (currently UG05), with or without the 23-byte data area mentioned above.
B. XCTL to the external initiation module (currently UGZ0008P), with or without the 23-byte data area mentioned above.

If no data area is passed, the effect is exactly the same as if a terminal level initiation of the guest occurred.

When a data area is passed, control-dispatch attempts to sign the user on to UMS using the ID/PASSWORD fields in the data-area. If this fails, the effect is the same as if no data had been passed.

If the signon succeeds, the passed entry-reason code is checked for being a code which could have been developed from keyboard entry. If it is not, it is forced to the code for ENTER.

The entry is then sent through function-dispatch exactly as if it had come from a keyboard. Generally, all functionality which would have been present from the UMS function selection screen is supported. The only current exception to this is that if the supplied entry-reason code references F4 or F9, any rescroll data present for the device is lost.

11. Scroll applications are expected to use the specified scroll-area for their functionality, and thus not interfere with applications that might have exited to them for duplicate resolution.

12. Each guest business function is expected to maintain its own view of the structure of the guest common work-area. There is no supported concept of a global view
of this area, and no assurance is thus present that all (or any) other applications will follow the convention used by one particular application for this area. This approach is essential for functional independence as well as the ability to test functions on a modular basis. The same general concept applies to the host common work-area.

Note that this discussion applies to areas described as application work areas, and NOT to areas described as owned by control-dispatch.

13. Clearly, the intent is that guest applications which specify a host interface area consider this area to be an area provided in addition to the work-area at the end of guest-common. Applications are expected to move data directly to and from this area, without unnecessary buffering through common. It has clearly been stated that if required, an ability to reference LXTBL processing to the host interface area will be provided, as well as control-dispatch table-driven maintenance of the host interface area. When the guest initiates some action, the host may take the view that the guest's host-interface area is present in the specified section which is in the middle of the host common. The actual amount of data moving in each direction is driven by the detail length fields on both the host and guest. It is important for performance reasons that the various host interface area formats be constructed such that the minimum amount of space be used. Under no circumstances should the application take the arbitrary attitude of always sending the maximum length.

14. Host applications should be constructed such that the minimum data needed for a given guest activity is returned to the guest via the host interface area. The passing of data elements (or data records) not needed for the current guest activity is not to occur. In a similar manner, the guest sends only minimum data to the host. For a typical retrieval operation, this would be perhaps a key and key-type. For a typical update operation, this would be surrogate(s) and only the new (or changed) data elements. In some critical update operations, also a significant data element (which would be examined by the host for change, inhibiting the operation if an intervening change occurred) might also be passed.

15. There is clear intent to push activity to the guest, as well as to control activity on the host. Host functions are intended to be data activities limited in scope and with necessary relationships to each other. This serves as a natural choke on the extent to which the various guests can load the host. In terms of the current system, for example, the points at which RS, LP, and RN say “READ LIMIT EXCEEDED” (or equivalent language) require a host return to guest. In a similar fashion, the design of the guest to host requests for code being written must be such that high host loads are avoided. Multiple guest to host interactions are preferable.
16. Message text for output should be through the message-code, message dictionary facility. In defining the messages, the alias feature should be used whenever possible.

17. The application work area in guest common is 2290 bytes in length. The application work area in host common is 512 bytes in length, it is preceded by the 2048 byte host interface area.

18. The general relationship between function-dispatch and associated applications is an XCTL relationship. In all cases, common (host or guest) is passed, using as a length the half-word length cell contained in the appropriate common-area. Applications MUST NOT have a hard-coded common length. The relationship between various modules described in a given PCTE entry may be XCTL or LINK (in either case passing common as above). The LINK relationship is not desirable, and if used should not go down but one level.

19. Modules (as opposed to functions) are not to be shared between UMS and ALAR. This is for several reasons including (but not limited to) a requirement for not having to test both systems in parallel during modification phases, differences in coding conventions/structures and significant differences in internal security structure.

20. UMS contains a call-by-address mechanism which maintains reentrancy on the called modules. This requires coding techniques which have not been applied to call-by-address modules within ALAR, and that allow for CICS releases which dispatch applications under multiple TCB’s thus requiring true reentrancy (not quasi-reentrancy). As guests may be (and often are) running under CICS releases well in advance of those at RMV, these techniques (and control-dispatch owning of the involved modules) is a requirement for this kind of structure.

21. The use of most of the Program Function Keys is standardized throughout the system. Deviations from the standard must be approved in advance to avoid adverse impact on the rest of the system and its users. The uses are listed on page 23.

At this point it would be helpful to read through the COBOL field names and the assembler comments on the Guest Common Area field-layouts in Appendix "F" and review the table beginning on page 27.
General Structure Specification

Host Side, “UMS” Interface

The host side of this product will consist of a set of modules which may be viewed as existing in 5 logical levels, but in practice run in a 2 level structure. The following diagram demonstrates the logical level structure, as best as can be represented in two dimensions:
Note that in the logical view, each level communicates down one level in a bidirectional manner. Also, note that levels 4A and 4B should be viewed as being at the same logical hierarchy, but so defined as to maintain a strict isolation between database and non-database functionality.

This structure is logically correct, but introduces a series of relationships which do not provide for the best utilization of resource under a CICS monitor. For this reason, the logically separate levels (1, 2, & 3) are really at the same physical level in the actual implementation. Each passes off laterally to its successor, and the last level 3 sequence invoked passes off to the completion phase of level 2, which in turn passes off to the return phase of level 1.

**Guest Side Conventions**

The general structure of the guest-side processes for UMS is similar to the existing ALAR structure in that there is a control-supervisor structure which serves to provide terminal services, a memory (common) area, interpretation of screen function-code, etc. A set of service functions is also provided. The control-dispatch software is referred to as “CD.”

The general flow of events is that CD will be entered as a result of a terminal input and will ascertain if an active process is in progress for the transaction. If not, the active process will be set to the signon process. In either case, a common-area will be acquired (of the proper length for the process), and seeded with a value-set. The value-set will be composed of values saved from prior invocations and information concerning the current invocation.

CD will transfer to the specified application process via XCTL, passing a commarea. This is the general pattern of operations throughout. CD seeds the commarea with a number of module names including that of a cleanup module to which control must pass when the application is complete. This process will issue the required transmission to the terminal.

CD passes the first application process a set of flags concerning the reason for entry. The potential reasons include (1) entry due to new function-code, (2) reentry due to enter, (3) reentry due to F7/8, (4) entry due to F4, (5) reentry after host-interface, etc. It is important to note that there is a high emphasis on small modules which use XCTL to pass off for a next logical phase, as opposed to large single modules or many level deep communication via LINK. The host interface requires access via XCTL.

The guest processing pattern is driven from a process control table. This table relates external function-codes with a set of program-names, specifies function key functionality/restrictions, defines common-area length, etc.
Because the intent is to provide code to the insurance industry which has reasonable potential for customization and device migration, it is important that the front-end function of input-map field fetching be separated from field processing. The guest common area structure is given in detail in another chapter. The structure provides for a scroll (browse) surrogate table and page table. Two bytes are provided for indicating information relative to the format of this table. We intend that a relationship between the process-control-table and the current format of these two bytes indicate if a current F4 type transfer is valid. Further, the F4 target is to be provided with information about its source based on this input. We have no firm position on if the meaning of these two bytes should algorithmically define the table, or if they should be a code defining the table.

A number of cells are provided for GETMAIN areas, including map-in (provided by CD), map-out, host-interface. Due to the requirement of complete input processing first, the map-out area could overlay the map-in area. Further, if this were desirable, the maximum map-out length for the process could be included in the process-control-table entry for the function as obtained by CD. The CD save-commarea function is used during the host interface activity.

**Host Side Conventions**

A major component of this development effort is the isolation of the various components of the system from each other. The major components relative to the HOST side are:

1. Protocol
2. Functional Control
3. Primary Security
4. Secondary Security
5. Primary Application Processes
6. Applications Service Routines
7. Database Service Routines

In order to maximize performance, every effort should be made to devise application modules as short, functional processes that complete their designated activity and XCTL to the next required functional process (or back to the cleanup process specified by function dispatch if the application activity is complete). When choices must be made relative to size, emphasis should be first on reducing working-storage and secondly on reducing procedural sizes. Clearly, coding conventions that assist in this process should be utilized. For example, switches and flags should be 1 byte alpha-numeric items and compared to (or set to) literals as opposed to data-items.

The goal of component isolation is best achieved in our current environment in the area of database access. The structure of this product implies that the database is sufficiently removed from the applications that the access method might be changeable without
application change. In turn, this means that database functionality would be served by a set of service modules which would accept function-codes (and involved data), and perform actions (returning status information and/or data). Thus, in the current environment, all IDMS verbs and interactions exist at the 4B (database service) level only. The parameter and data-record passing with the higher level(s) is done with standard CICS and COBOL mechanisms, meaning that the higher level modules are not compiled with the IDMS preprocessor and that standard copy members exist for the passage of data between the 4B and other levels.

In a similar fashion, most of the security processing is completely removed from the applications. The only extent to which the applications are involved with security concerns cases where the security determination is based on field content. For these situations, the application needs to be aware of the existence of a security constraint concerning a given field within the application. The responsibility of the application (when retrieving the involved data) is to call the security assist routine passing the address of the field(s) and an identifying function-code. The security assist routine will pass back a return-code from a set such as the following:

- 00 No security constraint implied.
- 01 User cannot see field. Security has overlaid field data with “nonexistent” value for passing to user.
- 02 User cannot see involved record. Treat identical to record-not-found.
- 03 User process to be abended. Security has seeded return area with appropriate error-code. XCTL to cleanup process.

It is important to note that when the applications traverse down levels using LINK (or CALL if an assist routine is involved), the error-code field in the common-area text body needs to be evaluated for a non-zero content. If this circumstance occurs, the invoked level must immediately return up 1 level. Since each level will contain such checking (similar to a return-code test) the effect will be that as soon as the error-code becomes non-zero, the cleanup process will be invoked.

**Inter-Process Communication, “UMS” Interface**

In the traditional fashion, a common area will be used to maintain the memory items necessary for any invocation of these processes. This common area will have several differences from the one in use for the existing ALAR Supervisor. The most glaring of these differences is that the common area will NOT be preserved across invocations from the same end node.

In concept, the UMS Protocol Processor will acquire the common area, initialize certain cells and pass the area to the Function Dispatch Processor. In turn, the area will be passed from process to process until the cycle is complete. The detail structure of the common area is given later. Its general structure is as follows:
Area 1: Protocol Control Area:
This area is reserved for use by the protocol processor.

Area 2: Module Control Name Area:
This area is used for the storage of module-names that are required in the course of the various processes. These names are placed here at transaction start-up time by the UMS-GUEST Control Program. The control program extracts these names from the Guest Program Control Table (PCTEG) based on the transaction code entered. The intention is that no program manipulate hard-coded program name literals. The literals will only be in the table so that a change in the table will be completely effective throughout the entire run-time system.

Area 3: Assist Routine/Table Addresses:
This area is used for the storage of the entry-point addresses of functional assistance modules which are commonly used and whose residence is insured prior to invocation of any application process. Typical examples of entries here are date-conversion routine, address conversion routine, specialized move routine for structuring return messages, etc.

Area 4: Post-Function Security Data:
This area is used to hold data required for security functions which cannot be evaluated prior to function initiation. The design of the UMS processing is that the function-dispatch level is cognizant of any items which the existing ALAR software would classify as subfunctions and interrogates security prior to entering applications on this level. However, this information is used for the resolution of data dependent security items.

Area 5: Interchange Text Body:
This area holds the message from/to the guest. The input-side protocol process seeds this area with the input text, and clears all cells used to control output/error flow. The applications (using the provided assist routines) build the response (and error-codes) in this area for eventual return to the guest. The defined size of this area will limit the maximum length response possible for a given message.

Area 6: Data Services Work Area:
This area is a generalized area used for both communications with the service routines (database and non-database), as well as a work-area for the invoked application programs. By convention, the service routines will use the area from the top down, and the applications will use the area from the bottom up.
Program Function Key Standards

The UMS System contains capabilities programmed into the function keys. Throughout the UMS Subsystem, some of these function keys can be used to move back and forth between primary and secondary sessions and functions, while others facilitate simpler operations, such as scrolling the screen data.

The following is an explanation of each function key's use within the system. Please note that this is a general list for the entire system, and not all function keys are available to each function. The individual documentation for each function will indicate the function keys available.

**F1**  The F1 key will return the user to the signon screen. However, if the user is currently positioned at the signon screen, the use of the F1 key will return to CICS.

**F2**  The F2 key moves the user to the UMS Menu screen. Use of this key is equivalent to entering UMM in the function field and pressing the “Enter” key.

**F3**  The F3 key can be used only in a primary session with screens containing a built-in reference list. The user must request a function change and move the cursor to a selectable item on the reference list before invoking the limited secondary session via F3.

**F4**  The F4 key is used to select a record from a scroll screen and switch to a secondary function. The user must enter the desired function and move the cursor to the desired record before invoking this function.

**F5**  The F5 key is not currently used in UMS.

**F6**  The F6 key is used to invoke the “Screen Hop” facility, a method of carrying information to another inquiry screen after a change of function code.

NOTE: As this procedure is unique and fairly involved, the user should consult the separate documentation on the "Screen Hop" facility for a complete explanation.

**F7**  The F7 key is used to page backwards on a scroll screen. This key is also used on screens which process more records than can fit on one screen.

NOTE: Although UMS allows paging forward through an entire set of screen data, backward scrolling permits a maximum of eleven (11) pages.
The F8 key is used to page forward on a scroll screen. This key is also used on screens which process more records than can fit on one screen.

The F9 key is used after UMS has automatically invoked a scroll screen due to the specification of a duplicate key by the user. The cursor is moved to the desired line item and F9 is pressed to return to the original function.

The F10 key is not currently used in UMS.

The F11 key is used to reinvoke the last scroll function at the exact point the user left it to invoke the current function. The user must enter the appropriate function code prior to pressing F11.

The F12 key is used in update functions to update the database with the values shown on the screen.

Special Session Definitions

Scroll Screens

Some screens are classified as scroll screens. This implies that they present a list of records whose keys are in some way similar (or identical). The screens may be entered by direct user entry of the function code, or automatically by the programs if the programs determine that the user entry is insufficient to decide between similar records stored on the database. The idea is that the user will, by means of cursor positioning and function keys, make the final determination of the desired record.

Most Scroll screens support paging forward and backward, as well as paging by partial pages. The actual selection of the desired record is accomplished by an internal “Browse Table” mechanism.

Reference Lists

Some screens build a “Reference List” to provide additional information about records on the screen. This list is accessed and used in a manner similar to the Scroll Screen/Browse Table mechanism, but is intended for brief examination of supportive detail rather than as a general processing path.
Primary and Secondary Sessions

UMS allows the user to maintain both the normal (or Primary) session and a Secondary session. When the secondary session is invoked, the entire environment which describes the primary session is preserved and restored on return. The intent is that the secondary session can be used for detail reference which may be required to continue the activity being accomplished in the primary session, without the necessity to back out of the detail entered in the primary, and later reenter it. Generally, any inquiry function can be accomplished in either session, but update functions are restricted to the primary session.

At any time, the user can determine which session is active by examining the date in the top left corner of the screen.

♦ If the separators are the “/” character, the session is primary.
♦ If the separators are the “-” character, the session is secondary.

The user may directly invoke a secondary session which allows all inquiry functions, or may cursor select a limited secondary session which allows only one inquiry function.

Guest-Host Control Blocks

The guest/host (guest-to-host and host-to-guest) blocks are used as communication areas between the guest software at the insurance companies, and the host software at the RMV. Guest-to-host blocks contain data to be sent from the guest to the host side, and host-to-guest blocks contain data to be sent from the host to the guest side.

These blocks are defined in the host and guest programs, and the host-to-guest block for a screen is usually defined in the LX table. An LX table is a macro that performs editing and mapping for a screen. If changes are made to the host-to-guest block, then the SE must make the changes to the applicable LX table also.

NOTE: The UMS screen “UPH” will be used as an example throughout this description.

The guest-to-host blocks contain key data fields and information. The key data fields, moved from the applicable UMS screen, that are necessary to perform the database retrieval on the host side. The other data in this block consists of data retrieved from the database and data used for updating screens. The guest-to-host block for each UMS screen is different because all screens have unique key data fields and some are inquiry only while others are updatable. Guest-to-host blocks vary in length, but each has 62 bytes at the top that are “reserved” for specific functions.

♦ The first 40 bytes of the 62 contain routing data, security data and the four-byte internal function code for that screen.
The next 22 bytes of the 62 contain error information, various indicators and other information, or may be used for duplicate key logic.

For “UPH,” these 22 bytes contain key data for the screen so that if duplicates are encountered during host data base retrieval, duplicate processing may be based on the contents of the 22 bytes.

Interface modules are invoked when transferring from the guest side to the host side and vice-versa. These interface modules use the first 62 bytes of the guest-to-host blocks for their processing (routing and duplicate key logic). The interface modules send the 62 bytes, along with the rest of the block, to the applicable host program in its common area.

Each guest input program moves the length of its guest-to-host block to the common area before transferring control to the host side. This length tells the interface modules the length of the applicable guest-to-host block being passed. The length is determined by subtracting 40 bytes from the total length of the guest-to-host block. The 40 bytes subtracted are part of the 62 reserved bytes at the beginning of the block.

The “UPH” screen’s guest-to-host block consists of 40 bytes used for routing, security, and the internal function code, followed by 22 bytes used for duplicate key logic, followed by the rest of the block. The total length of this guest-to-host block is 214 bytes, so the response length is 174 bytes:

\[ 214 \text{ bytes} - 40 \text{ bytes} = 174 \text{ bytes}. \]

When NDR was implemented, the “UPH” screen’s 22 bytes, reserved for duplicate key logic, were not enough to hold the 25 byte out-of-state license plus the 2 byte state code (27 bytes total). Five bytes were added to the block, immediately following the 22 bytes, to hold the full 27 bytes needed for duplicate key logic.

Once control has been sent to the host side, the applicable host program saves the guest-to-host block received by moving it from one area of the common area to another. The applicable host program is either the input that performs data base retrieval, or the output that performs database updates. The area that the guest-to-host block had previously occupied is initialized to low-values and will be filled with the host-to-guest block to be sent back to the guest output program. The host-to-guest block will contain data moved from the guest-to-host block, in addition to applicable data obtained after the host program has performed the database retrieval/update. The majority of the data sent back to the guest output program, in the host-to-guest block, will be displayed on the screen.

Before transferring to the guest side, the host program moves the length of its host-to-guest block to the common area. This length will tell the interface modules the length of the applicable host-to-guest block being passed. Each host-to-guest block has a different length. For the “UPH” screen, the block consists of a fixed header (174 bytes) and up to ten details (each is 47 bytes). The length may vary based on how much data was retrieved from the database (in contrast to the “UPH” screen’s guest-to-host block, which
is constant). The formula used to compute the length for the “UPH” screen's host-to-guest block is as follows:

\[ 174 + (47 \times \text{# of details}) = \text{length} \text{ (up to 644 bytes)} \]

When the guest output program receives control from the host program, it will process the data transmitted in the host-to-guest block. Some of this data is moved to the common area. When the LX table is executed, data from the common area is moved to the output map. When the guest input program receives control from CICS again, it will take the data from the common area (which the guest output program moved), plus any new data entered on the screen, and move it to the guest-to-host block. This block will again be sent to the host programs and the cycle repeats itself.

**Guest Common Area Structure**

(4074 bytes in length)

Note that each section of the common-area has been described in a separate definition. The length of each section is given, and the content is described as offsets relative to the section. COBOL and BAL layouts will be provided. Areas which are used by high-level functions (and are transparent to the applications) have not been detailed. Most areas preceding the application work-area must be treated as read-only by the applications. There is a small set of exceptions to this which are noted. The general organization of this structure is very similar to the host side.

**Protocol Control Area** (64 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0007</td>
<td>08</td>
<td>general purpose double-word aligned temporary</td>
</tr>
<tr>
<td>0008-0009</td>
<td>02</td>
<td>length of common-area. halfword aligned for XCTL, LINK.</td>
</tr>
<tr>
<td>0010-0011</td>
<td>02</td>
<td>current inquiry/response detail length. initialized to zero and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>maintained by appropriate assist routine(s).</td>
</tr>
<tr>
<td>0012-0015</td>
<td>04</td>
<td>current function-name</td>
</tr>
<tr>
<td>0016-0017</td>
<td>02</td>
<td>internal control flag bits</td>
</tr>
<tr>
<td>0018-0021</td>
<td>04</td>
<td>previous function-name</td>
</tr>
<tr>
<td>0022-0023</td>
<td>02</td>
<td>length of last receive-map</td>
</tr>
<tr>
<td>0024-0035</td>
<td>12</td>
<td>process control flags from PCTE for current function</td>
</tr>
<tr>
<td>0036-0042</td>
<td>07</td>
<td>previous map-name</td>
</tr>
<tr>
<td>0043-0043</td>
<td>01</td>
<td>internal flag</td>
</tr>
<tr>
<td>0044-0047</td>
<td>04</td>
<td>next or current transid</td>
</tr>
<tr>
<td>0048-0048</td>
<td>01</td>
<td>internal flag</td>
</tr>
<tr>
<td>0049-0063</td>
<td>15</td>
<td>reserved for protocol dependency</td>
</tr>
</tbody>
</table>
### Module Control Name Area

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0007</td>
<td>08</td>
<td>guest-to-host protocol interface module</td>
</tr>
<tr>
<td>0008-0015</td>
<td>08</td>
<td>1st application service module for current invocation</td>
</tr>
<tr>
<td>0016-0023</td>
<td>08</td>
<td>2nd application service module for current invocation</td>
</tr>
<tr>
<td>0024-0031</td>
<td>08</td>
<td>3rd application service module for current invocation</td>
</tr>
<tr>
<td>0032-0039</td>
<td>08</td>
<td>4th application service module for current invocation (cleanup module)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>note that only 1 application module is required for a given function.</td>
</tr>
<tr>
<td>0040-0047</td>
<td>08</td>
<td>after-host return module. copied by application into this cell from one of the preceding 4.</td>
</tr>
<tr>
<td>0048-0055</td>
<td>08</td>
<td>optional edit/map/demap table (LXTBL) module</td>
</tr>
<tr>
<td>0056-0063</td>
<td>08</td>
<td>crt sendback module</td>
</tr>
<tr>
<td>0064-0071</td>
<td>08</td>
<td>internal dispatch routine</td>
</tr>
<tr>
<td>0072-0079</td>
<td>08</td>
<td>reference list manager</td>
</tr>
<tr>
<td>0080-0087</td>
<td>08</td>
<td>non-resident services module</td>
</tr>
<tr>
<td>0088-0095</td>
<td>08</td>
<td>Lxtable processor</td>
</tr>
<tr>
<td>0096-0103</td>
<td>08</td>
<td>report print proc</td>
</tr>
<tr>
<td>0104-0111</td>
<td>08</td>
<td>level 2 host call</td>
</tr>
<tr>
<td>0112-0119</td>
<td>08</td>
<td>CICS BR14 equiv</td>
</tr>
<tr>
<td>0120-0127</td>
<td>08</td>
<td>host IO handler</td>
</tr>
<tr>
<td>0128-0135</td>
<td>08</td>
<td>printer interface</td>
</tr>
<tr>
<td>0136-0191</td>
<td>56</td>
<td>room for additional names</td>
</tr>
</tbody>
</table>

### Assist Routine/Table Area

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0003</td>
<td>04</td>
<td>address of date-conversion module</td>
</tr>
<tr>
<td>0004-0007</td>
<td>04</td>
<td>address of data-name to address conversion routine</td>
</tr>
<tr>
<td>0008-0011</td>
<td>04</td>
<td>address of text fetch routine for host output text</td>
</tr>
<tr>
<td>0012-0015</td>
<td>04</td>
<td>address of text fetch routine for host input text</td>
</tr>
<tr>
<td>0016-0019</td>
<td>04</td>
<td>address of guest control-table</td>
</tr>
<tr>
<td>0020-0023</td>
<td>04</td>
<td>address of translate/test table set</td>
</tr>
<tr>
<td>0024-0027</td>
<td>04</td>
<td>address of edit utility set</td>
</tr>
<tr>
<td>0028-0031</td>
<td>04</td>
<td>address of Shell Sort routine</td>
</tr>
<tr>
<td>0032-0191</td>
<td>160</td>
<td>room for 40 additional entries</td>
</tr>
</tbody>
</table>

### Post Function Security Data

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0023</td>
<td>24</td>
<td>reserved for security interface data  (5 addresses used, 1 available)</td>
</tr>
</tbody>
</table>

---

Chapter Two - Control - Dispatch
Scroll Function Data  (742 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0003</td>
<td>04</td>
<td>last scroll-function name</td>
</tr>
<tr>
<td>0004-0005</td>
<td>02</td>
<td>reserved</td>
</tr>
<tr>
<td>0006-0017</td>
<td>12</td>
<td>scroll table descriptor. built by application program. 1 byte #entries per line, zoned decimal. 1 byte #surrogates per entry, zoned decimal (1-8) 2 bytes first line used on screen, binary 8 bytes surrogate type table</td>
</tr>
<tr>
<td>0018-0084</td>
<td>67</td>
<td>scroll key. built by application program.</td>
</tr>
<tr>
<td>0085-0085</td>
<td>01</td>
<td>core status flag. set by application. must be ‘Y’ if scroll-area is considered valid.</td>
</tr>
<tr>
<td>0086-0181</td>
<td>96</td>
<td>page management table. Used by application for page forward and page back. Format is at application discretion. (12 8-byte entries)</td>
</tr>
<tr>
<td>0182-0741</td>
<td>560</td>
<td>scroll/browse table. set of up to 140 4-byte entries surrogate entries formatted as indicated by the scroll descriptor table. Built by application, used by function-dispatch.</td>
</tr>
</tbody>
</table>

WARNING: Cells above this point are read-only unless otherwise noted. Cells below this point are read-write unless otherwise noted.

Data Services Work Area  (512 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0001</td>
<td>02</td>
<td>flags for data-move routines</td>
</tr>
<tr>
<td>0002-0021</td>
<td>20</td>
<td>date conversion area. contains EIBDATE in binary format on initial function-entry if LXTBL is not used.</td>
</tr>
<tr>
<td>0022-0028</td>
<td>07</td>
<td>name of current map. set by application if not using LXTBL. Read-only if using LXTBL.</td>
</tr>
<tr>
<td>0029-0030</td>
<td>02</td>
<td>Entry reason-code for current invocation</td>
</tr>
<tr>
<td>0031-0031</td>
<td>01</td>
<td>type of surrogate if entry is F4 or F9. read-only.</td>
</tr>
<tr>
<td>0032-0035</td>
<td>04</td>
<td>surrogate value if entry is F4 or F9. read-only.</td>
</tr>
<tr>
<td>0036-0037</td>
<td>02</td>
<td>EIBDATE in binary (WAASDATE) format. read-only.</td>
</tr>
<tr>
<td>0038-0097</td>
<td>60</td>
<td>reserved surrogate list. application agreement is required on which significant surrogate is in which cell.</td>
</tr>
<tr>
<td>0098-0101</td>
<td>04</td>
<td>output map-length. read-only if LXTBL used.</td>
</tr>
<tr>
<td>0102-0105</td>
<td>04</td>
<td>address of input-map area. read-only.</td>
</tr>
<tr>
<td>0106-0109</td>
<td>04</td>
<td>address of output-map area. read-only.</td>
</tr>
<tr>
<td>0110-0113</td>
<td>04</td>
<td>address of host-interface area, zero if no area. read-only.</td>
</tr>
<tr>
<td>0114-0129</td>
<td>16</td>
<td>reserved for up to 4 additional addresses. read-only.</td>
</tr>
<tr>
<td>0130-0130</td>
<td>01</td>
<td>error-intercept flag. set to ‘Y’ by application if host errors are to be intercepted without return to the application.</td>
</tr>
<tr>
<td>0131-0131</td>
<td>01</td>
<td>internal dispatch type. application must set to ‘0’ if using an internal dispatch service.</td>
</tr>
<tr>
<td>0132-0135</td>
<td>04</td>
<td>name of function requested for internal dispatch.</td>
</tr>
</tbody>
</table>
0136-0159  24  first 24 detail bytes from last host call.  read-only.
0160-0511  352  reserved

Undefined Area  (2 bytes)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0001</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Error/Message Area  (54 bytes)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0003</td>
<td>04</td>
<td>current error/message code</td>
</tr>
<tr>
<td>0004-0053</td>
<td>50</td>
<td>override text</td>
</tr>
</tbody>
</table>

Cursor Area  (2 bytes)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0001</td>
<td>02</td>
<td>cursor position data.  read-only if LXTBL used.</td>
</tr>
</tbody>
</table>

Individual Application Work Area  (2290 bytes)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-2289</td>
<td>2290</td>
<td>reserved for any use any application wants to make of this area. Preserved across psuedo-conversational interactions by control-dispatch. not guaranteed relative to format between applications.</td>
</tr>
</tbody>
</table>

UMS Software-Interface Hooks

The guest side of the UMS system has been designed to minimize points of interface with the external environment and to provide specific points where a user site can hook to the system with reasonable levels of effort and impact.

As a general design, UMS Guest:

♦ Interfaces with the host in only one module - UGZ011P
♦ Obtains normal 3270 input in only one module - UGZ0001P
♦ Outputs normal 3270 output in only one module - UGZ0005P

Normal 3270 input/output is used in the context of CICS RECEIVE-MAP and SEND-MAP. Additionally, the 3270 device is accessed in READ-BUFFER mode and/or SEND mode when the secondary session is entered or exited.
As a design objective, we have assumed that the user site might typically desire to log transmissions to and/or from the host and/or generate its own interactions with the host.

To this end, the host activity was centered in one module, UHZ0011P (not distributed). We have tried to make it relatively easy for the user to write these activities to a log or invoke a user program (exit) to process them. To that end, we wrote the code such that we believed the points at which this would be done to be relatively obvious.

Secondly, the data structure passed to UGZ0011P is such that it can be generated relatively easily without using all of the UMS Guest software. This allows invoking the host from some other program at the guest site without using the normal guest dispatch-control mechanism. As a matter of practice, this is the technique used by UGTL/UGTH (the guest initialization) transactions to invoke the host and can be seen in a limited form in UGZ0020P.

If a user site elects this approach, the site commits itself to using the RMV’s guest host data block formats. These formats are subject to potential change, but have been designed to make the upgrade as painless as possible. In particular, the design includes a 4 byte block type field with a commitment on the part of the RMV to provide an overlap period where both the old and new block type codes function. Such an overlap period allows the unmodified guest an upgrade window and the modified guest a development window (if required). Because the RMV is committed to at least one guest which is totally a user development effort, it is clear that some degree of documentation will proceed any block type changes.

It would also be possible for a user site to hook into the UGZ0001P and UGZ0005P programs with the intent of simulating BMS map input/output functions. However, the following cautions are in order:

1. Across releases to date, UGZ0011P has been very stable and we expect this to be an ongoing trend. UGZ0001P has not been stable, and we do not expect it to be. UGZ0005P has been more stable than UGZ0001P and less than UGZ0011P. Any user hooking these programs commits themselves to perpetuating their code across all new releases in critical modules where the RMV has no design intent to support release exits or other assistance for this effort.

2. The RMV has every intention of modifying (as required for bugs or development) screen formats at release time. Such modifications would seriously impact any user who elected to hook in this method.

3. The current UMS Guest takes some effort to be reasonably efficient in its 3270 transmissions. Such efforts will certainly increase as the product matures. Knowledge of this position should indicate to anyone planning to hook our SEND-MAP images the amount of work required. Also any such effort would have to at least be cognizant of the 3270 READ-BUFFER usage.
Host Common Area Structure

(3072 bytes in length)

Note that each section of the common-area has been described in a separate definition. The length of each section is given, and the content is described as offsets relative to the section. COBOL and BAL layouts will be provided. Areas which are used by high-level functions (and are transparent to the applications) have not been detailed.

**Protocol Control Area**  (64 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0007</td>
<td>08</td>
<td>general purpose double-word aligned temporary</td>
</tr>
<tr>
<td>0008-0009</td>
<td>02</td>
<td>length of common-area. halfword aligned for XCTL, LINK.</td>
</tr>
<tr>
<td>0010-0011</td>
<td>02</td>
<td>current response detail length. initialized to zero and maintained by appropriate assist routine(s).</td>
</tr>
<tr>
<td>0012-0071</td>
<td>60</td>
<td>reserved for protocol dependency</td>
</tr>
</tbody>
</table>

**Module Control Name Area**  (192 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0007</td>
<td>08</td>
<td>output-side protocol processor name</td>
</tr>
<tr>
<td>0008-0015</td>
<td>08</td>
<td>cleanup processor name</td>
</tr>
<tr>
<td>0016-0023</td>
<td>08</td>
<td>1st application service module for current invocation</td>
</tr>
<tr>
<td>0016-0191</td>
<td>176</td>
<td>room for 22 additional names (13 in use)</td>
</tr>
</tbody>
</table>

**Assist Routine/Table Area**  (192 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0003</td>
<td>04</td>
<td>address of date-conversion module</td>
</tr>
<tr>
<td>0004-0007</td>
<td>04</td>
<td>address of data-name to address conversion routine</td>
</tr>
<tr>
<td>0008-0011</td>
<td>04</td>
<td>address of text insertion routine for output text</td>
</tr>
<tr>
<td>0012-0015</td>
<td>04</td>
<td>address of data-dependent security evaluation routine</td>
</tr>
<tr>
<td>0016-0191</td>
<td>176</td>
<td>room for 44 additional entries</td>
</tr>
</tbody>
</table>

**Post Function Security Data**  (64 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0063</td>
<td>64</td>
<td>reserved for data-dependent security function use</td>
</tr>
</tbody>
</table>
### Interchange Text Body

(2048 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0023</td>
<td>24</td>
<td>Routing information. Used by Guest. Returned with response.</td>
</tr>
<tr>
<td>0024-0035</td>
<td>12</td>
<td>inbound security descriptor area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes guest site name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes guest sub-site name (agent)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes guest sub-site password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(note that this field is not valid when passed to application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>processes)</td>
</tr>
<tr>
<td>0024-0035</td>
<td>12</td>
<td>outbound timing data (provided by protocol processor)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 bytes same as inbound security descriptor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 bytes time of response</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 bytes date of response</td>
</tr>
<tr>
<td>0036-0039</td>
<td>04</td>
<td>Block (transaction) type code (same as inbound)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 bytes subsystem code (reg, lic, ums, etc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 bytes subsystem function code</td>
</tr>
<tr>
<td>0040-2047</td>
<td>2008</td>
<td>This area constitutes the maximum length variable text for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>guest-to-host or host-to-guest transmissions. The treatment of this</td>
</tr>
<tr>
<td></td>
<td></td>
<td>area is different depending on the direction of the transmission.</td>
</tr>
</tbody>
</table>

### Host-to-Guest

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0040-0043</td>
<td>04</td>
<td>Application dependent response status flags. Byte 0040 used for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scroll response:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘1’ data presented ends scroll (end-of-set)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘2’ data presented does not end scroll</td>
</tr>
<tr>
<td>0044-0047</td>
<td>04</td>
<td>error-code. from defined list.</td>
</tr>
<tr>
<td>0048-0049</td>
<td>02</td>
<td>response count. 00=none, 01=one, etc.</td>
</tr>
<tr>
<td>0050-0061</td>
<td>12</td>
<td>error qualifier for output.</td>
</tr>
<tr>
<td>0061-2047</td>
<td>??</td>
<td>transaction dependent response for output.</td>
</tr>
</tbody>
</table>

### Guest-to-Host

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0040-0060</td>
<td>22</td>
<td>preserved header portion of input transaction</td>
</tr>
<tr>
<td>0061-2047</td>
<td>??</td>
<td>variable text body</td>
</tr>
</tbody>
</table>

The concept being suggested here is that logically the variable portion of the input text body consist of two areas. These areas are separate only in that the function dispatch process will isolate the first 22 bytes previous to invoking the application process, and place it as shown below. From an application perspective, this allows formatting the inbound text with critical data first, and so placed that the application need not have concern about overlaying this data with initial response data. From a control overhead
perspective, in the case of longer input text bodies, the time spent shifting data down (as was previously envisioned) is reduced.

**Data Services Work Area** (512 bytes in length)

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Len</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0511</td>
<td>512</td>
<td>reserved for data-services and application use. First 22 bytes are used to preserve the first 22 bytes of the input text body. Next 16 bytes are defined as standard parameter locations for assist routines. When not calling the routines using these cells, they may be overlaid.</td>
</tr>
</tbody>
</table>

**Verify Program (Batch) Sample Report**

Beginning on the next page is a Batch Sample Report.
Verify Program (Batch) Sample Report

Note: This example has been modified by text-edit to improve its readability and to shorten some of the sections.

UMS-GUEST LOAD MODULE ANALYSIS

DATE: 91.196 TIME: 14.05.50 PAGE: 00001

MODULE: UGZ0014P BAL_TABL BASELINE LKED-90287 LEN-010232
UGZ0014P: TYPE=BOOTH _CLEN=DFLT PROTOCOL=UGZ0013P SYSTEM=XA SITE=MRMV HOSTNAME=MVS1 HOSTTRAN=UH62
HOTKEY=PA1 COMSAVE= TRAN1=UG03 TRAN2=UG04 (E4C7F0P4) TEMPSOR=AUX
REFLIST=MAIN TTABLE=MAIN TRACING=NO #TENT=02034 NSWAP=01
TERM=SEQG 0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013 0014 0015
0016 0017 0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031
0032 0033 0034 0035 0036 0037 0038 0039 0040 0041 0042 0043 0044 0045 0046 0047
0048 0049 0050 0051 0052 0053 0054 0055 0056 0057 0058 0059 0060 0061 0062 0063
0064 0065 0066 0067 0068 0069 0070 0071 0072 0073 0074 0075 0076 0077 0078 0079
0080 0081 0082 0083 0084 0085 0086 0087 0088 0089 0090 0091 0092 0093 0094 0095
0096 0097 0098 0099 0200 0201 0202 0203 0204 0205 0206 0207 0208 0209 0210 0211

Note: If the program has a version 'stamp', as it should, it will be given in the report line otherwise, it will say 'BASELINE'.

MODULE: UGZ00002P BAL_TABL VER 03.01 LKED-91175 LEN-002968
MODULE: UGB0010T LXT_HDR LXT_BDY VER 01.00 LKED-91157 LEN-001008
MODULE: UGL0320T LXT_HDR LXT_BDY BASELINE LKED-91028 LEN-000552
MODULE: UGB0010M MAP_HDR MAP_BDY VER 01.00 LKED-91172 LEN-001200
MODULE: UGB0010P CBL_HDR CBL_BDY VER 01.00 LKED-91196 LEN-001296
MODULE: UGL0300P BAL_HDR BAL_BDY BASELINE LKED-91028 LEN-002048
NOTE: While a program marked >ERROR< may run 'OK' it will not be reliable in all environments. The condition must be resolved.

**UGRI110P COMPILED WITH DYNAM AND/OR RES**
**MODULE: UGRI110P CBL_HDR CBL_BDY BASELINE LKED-89244 LEN-002848 >ERROR<**

**MODULE: UGZ0003P BAL_TABL BASELINE LKED-90291 LEN-004552**

NOTE: The following report section, Host Functions W/ referencing
Guest functions will only be produced in a HOST-GUEST environment.

<table>
<thead>
<tr>
<th>HOST FUNCTIONS WITH REFERENCING GUEST FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EB01</td>
</tr>
<tr>
<td>EB02</td>
</tr>
<tr>
<td>HTB0</td>
</tr>
<tr>
<td>HTB1</td>
</tr>
<tr>
<td>HTB2</td>
</tr>
<tr>
<td>LB10</td>
</tr>
<tr>
<td>LD04</td>
</tr>
<tr>
<td>LD08</td>
</tr>
<tr>
<td>LI01</td>
</tr>
<tr>
<td>LI02</td>
</tr>
<tr>
<td>LI03</td>
</tr>
<tr>
<td>LI04</td>
</tr>
<tr>
<td>LI07</td>
</tr>
<tr>
<td>LI08</td>
</tr>
</tbody>
</table>
### Host Functions with Referencing Guest Programs

<table>
<thead>
<tr>
<th>EB01</th>
<th>UGB0010P</th>
<th>UGB0011P</th>
<th>UGB0020P</th>
<th>UGB0021P</th>
<th>UGB0022P</th>
<th>UGB0040P</th>
</tr>
</thead>
<tbody>
<tr>
<td>EB02</td>
<td>UGB0010P</td>
<td>UGB0011P</td>
<td>UGB0020P</td>
<td>UGB0021P</td>
<td>UGB0022P</td>
<td>UGB0031P</td>
</tr>
<tr>
<td>HTB0</td>
<td>UGB0020P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTB1</td>
<td>UGB0020P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTB2</td>
<td>UGB0020P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LB10</td>
<td>UGB0011P</td>
<td>UGL0260P</td>
<td>UGL0261P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD04</td>
<td>UGL0220P</td>
<td>UGB0002P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD08</td>
<td>UGL0240P</td>
<td>UGB0002P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Guest Programs with Referenced Host Functions

<table>
<thead>
<tr>
<th>UGB0010P</th>
<th>EB01</th>
<th>EB02</th>
<th>LI07</th>
<th>LI09</th>
<th>LI10</th>
</tr>
</thead>
<tbody>
<tr>
<td>UGB0011P</td>
<td>EB01</td>
<td>EB02</td>
<td>LB10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGB0020P</td>
<td>EB01</td>
<td>EB02</td>
<td></td>
<td>LI10</td>
<td></td>
</tr>
<tr>
<td>UGB0021P</td>
<td>EB01</td>
<td>EB02</td>
<td>LI07</td>
<td>LI09</td>
<td>LI10</td>
</tr>
<tr>
<td>UGB0022P</td>
<td>EB01</td>
<td>LI10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGB0030P</td>
<td>EB02</td>
<td>LI07</td>
<td>LI09</td>
<td>LI10</td>
<td></td>
</tr>
<tr>
<td>UGB0031P</td>
<td>EB02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGB0040P</td>
<td>EB01</td>
<td>LI01</td>
<td>LI02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0010P</td>
<td>LI01</td>
<td>LI02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0020P</td>
<td>LI04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0030P</td>
<td>LI03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0060P</td>
<td>LH10</td>
<td>LH20</td>
<td>LH30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0070P</td>
<td>LI07</td>
<td>LI08</td>
<td>LI09</td>
<td>LI10</td>
<td></td>
</tr>
<tr>
<td>UGL0200P</td>
<td>LI07</td>
<td>LI08</td>
<td>LI09</td>
<td>LI10</td>
<td>LX10</td>
</tr>
<tr>
<td>UGL0201P</td>
<td>LX20</td>
<td>LX40</td>
<td>LX50</td>
<td></td>
<td>LX20</td>
</tr>
<tr>
<td>UGL0220P</td>
<td>LD04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0240P</td>
<td>LD08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0260P</td>
<td>LB10</td>
<td>LI07</td>
<td>LI08</td>
<td>LI09</td>
<td>LI10</td>
</tr>
<tr>
<td>UGL0261P</td>
<td>LB10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGL0020P</td>
<td>LD04</td>
<td>LI04</td>
<td>UR02</td>
<td>UR03</td>
<td>HTB5</td>
</tr>
<tr>
<td>UGL0020P</td>
<td>HTB0</td>
<td>HTB1</td>
<td>HTB2</td>
<td>HTB3</td>
<td>HTB4</td>
</tr>
</tbody>
</table>
### Origin of Values

<table>
<thead>
<tr>
<th>Entries Used in Module/Table</th>
<th>Derived from Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>136</code> entries</td>
<td>As derived from the programs</td>
</tr>
<tr>
<td><code>86</code> entries</td>
<td></td>
</tr>
<tr>
<td><code>500</code> Assembled Size of Module/Function-Table</td>
<td>From MAXLOAD EQU 500 in assembly</td>
</tr>
<tr>
<td><code>146</code> entries</td>
<td>As derived from the programs</td>
</tr>
<tr>
<td><code>2000</code> Assembled Size of Host Function-Table</td>
<td>From MAX_REF_ENT EQU 2000 in assembly</td>
</tr>
<tr>
<td><code>85</code> Defined Host Functions</td>
<td>As derived from the programs</td>
</tr>
<tr>
<td><code>9</code> Fatal Errors Detected</td>
<td>&lt; Remember now, no fatal errors! &gt;</td>
</tr>
</tbody>
</table>

---

**End of Data**
Verify Batch - Sample Run JCL

// XXXXXX JOB < job card parms for your installation >

/// * ***************************************
/// *     BATCH UMS VERIFY PROGRAM       *
/// * ***************************************
///
///STEPLIB   DD DSN=< the whatever lib >,DISP=SHR
///SYSOUT    DD *
///DFHRPL    DD DSN=< lib with UMS modules >,DISP=SHR
///        DD DSN=< hopefully only one >,disp=SHR
All of the UMS application material (Subschemas, Copy Members, Programs, LX tables, and Maps) will use the following naming standard. The Module Name has the form of ‘ABCDDDE’ where:

- **A**: Constant of ‘U’ For UMS System
- **B**: Constant of ‘G’ For Guest Site
  - Constant of ‘H’ For Host Site
  - Constant of ‘I’ For Independent (used in both)
  - Constant of ‘V’ For Virtual Guests
- **C**: Application Function
  - B Booking System (exam)
  - C Cash
  - E Emmissions
  - H MAB
  - I Inspection Maintenance
  - L License Modules
  - M MRB
  - N Non-Renew License
  - P Policy Modules
  - R Registration Modules
  - S Suspension
  - U UMS & Cross System
  - V Motor Voter
  - W Overweight Permits
  - Z System Control Modules, General (edits, tables, etc.)
- **DDDD**: Number 1 through 9999
- **E**: Type of Program
  - P Program
  - M Map - (Guest side)
  - L Schemaschema - (Host side)
  - T LX table - (Guest side)
  - J JCL
G Copybook
Y Copybook
The UMS online subsystem uses different source, load, and macro libraries than ALARS. Following is a list of the libraries and their corresponding CVs:

### Source libraries:

- RMVMV.RMV.UMSSRCE CV10
- RMVMV.RMV.UMSSRCE2 CV13
- RMVMV.RMV.UMSSRCE3 CV14
- RMVMV.RMV.UMSSRCEP CV11

### Load libraries:

- RMVMV.RMV.UMSLOAD CV10
- RMVMV.RMV.UMSLOAD2 CV13
- RMVMV.RMV.UMSLOAD3 CV14
- RMVMV.RMV.UMSLOADP CV11

### Macro libraries:

- RMVMV.RMV.UMSMACRO CV10
- RMVMV.RMV.UMSMACR2 CV13
- RMVMV.RMV.UMSMACR3 CV14
- RMVMV.RMV.UMSMACRP CV11

When a programmer makes changes to a program or copy member, he/she should first look in the CV that has the current source. For example, if the source to be changed is found in CV14 and CV11, the source from CV14 should be moved to CV10. The changes should be made in CV10.

The CV11 production program-source, object modules, and macro source from production are loaded onto a tape and installed at each client site.
Proposed Module Movement Changes

This paper is a proposal to simplify the module movement and to combine the module movement of ALARS and UMS into one process. This document will address the movement of source, macro, and load libraries, but will not address the issue of subschema moves.

Currently the ALARS process is partly manual and partly automated. The UMS is completely automated. Both systems have separate sets of libraries. This is just a couple of many differences and possible areas for problems.

With the new library structure all of the systems will still have its own set of libraries. All modules will move to CV14 before going into production. This would give the users a chance to test the changes in an environment similar to CV11. Separate macro/copybook libraries will be maintained for the ALARS and UMS systems so that we are able to extract and deliver the needed members for the UMS guest sites. UMS will have an additional separation for the Host and Guest software.

An online (TSO) clist would be developed to enter all requests and generate the appropriate control cards for the movement of modules. Another series of reports would be developed to produce control totals and modules that are moving. All of this would eliminate the need for retyping of requests and any potential problems. Module movements would still occur on the days that are now in place. At a later point in time maybe Info Management might be the place to enter these requests.

While reading this document, keep in mind the drawings attached to get a feel of what is happening in this new scenario. You will notice that ALL library names have changed to better identify which IDMS region that they are working with. As today all developments/enhancements will be done in CV10.

After the tests have run successfully and it is desired to move the modules closer to production, a request can be entered to move the module(s) to CV14 and CV13. The modules will be moved, but this time to the HOLD libraries for ALARS MMS moves; UMS modules will be moved from CV10 to CV14 directly. The scheduled job will move these modules to the processing CV14 and CV13 libraries.

MMS and UMS modules, except MMS load, will be moved to appropriate production libraries. All MMS loads are moved to temp library RMVOL.JMV.TMPLIB, and after 10 days will be moved to the regular production library. On-line application MMS and UMS loads are moved to RMVMV.RMV.PRODLOAD library also.

Just a reminder that the modules are moved and not copied, therefore, the “from” library modules are deleted and can only be found in the “to” library.
The schedule of events is:

Every night except Saturday and Sunday - moves for CV13 and CV14
Tuesday nights - moves for CV11

If a module needs to move from CV10 to CV14 in the same night, it is done with a T to M (test to mirror) move procedure.

**UMS Move Clist**

1 **Overview**

A new procedure has been created to make the process of moving modules and copybooks between the various test regions and production simpler and hopefully less prone to error.

This process is now completely automated. All that a programmer needs to do to move programs and copybooks between the various regions is to request the move using the UMSMOVE clist.

Moves between all the “test” regions are scheduled to run Monday through Friday. Production moves are only scheduled to run on Tuesday nights, for Wednesday morning implementation.

Moves outside of these times will be emergency moves. That means that the move will be run straight away and programs will be “new copied” into CICS. To perform an emergency move, the request is made in the same fashion. A phone call should then be made to the Software Development Manager to get the move procedure completed.

If there are any problems with the clist, the Software Development Manager should be contacted.

2 **Procedure**

2.1 **Introduction**

There are two combinations of libraries that serve as origin and destination libraries. These are:

- TEST to MIRR
- MIRR to PROD
This enforces the rule that everything has to be moved to the mirror regions before going to production. If it is necessary to move entities from TEST or SPCL to production without doing any intermediate testing, then a move first needs to be made to mirror, followed in short succession by the move from mirror to production.

The programmer is responsible for specifying the members to move and the library combination. This is achieved using CLIST “UMSMOVE.”

The actual move will be performed by jobs scheduled to run in the evening. Production move jobs will run Tuesday night only for Wednesday morning implementation. All other moves will run Monday through Friday during the evening.

### 2.2 “UMSMOVE” Clist

“UMSMOVE” uses SPF panels to allow the programmer to specify:

- The origin/destination combination
- The names of members to be moved
- The type of member (program[P] or copybook/macro[M]).
- Date the move is to be performed
- Optionally a CSR number.

The SPF panels have help panels.

When the CLIST first starts up, it will display a panel asking the programmer to enter the Origin and Destination combination. Once entered, the CLIST will display another panel that will show any move entries outstanding for the programmer. The programmer can then add, delete, or modify the entries on the list.

Line commands “I” and “D” are available to Insert and Delete member entries. Updates to line entries are made by over-typing.

Validation is as follows:

- Member names are verified to ensure that they conform to IBM standards and that they exist on the appropriate library(ies). A program move assumes both source and load is to be moved.
- Member type - “P” for program (source and load), “M” for Copybook or Macro, “S” for Subschema source or program source only, “L” for program load only.
- Dates are verified to ensure they are valid (February 29th is not valid for this CLIST). It is important to use the correct use date. The date should reflect the next scheduled move or the actual date for an emergency. All entries that have passed a date, without being moved, will be deleted.
F8 and F7 can be used to scroll through the list of entries.

Command line entries are:

- **CANCEL** - Cancel changes and exit CLIST
- **SAVE** - Save changes made and continue with CLIST
- **END** - Save changes and exit CLIST
- **ALL** - Display entries for all users. This will only work for staff responsible for moves
- **NOTALL** - If “ALL” or “HISTALL” was entered, reverts to only display entries for person signed on.
- **H, HIST** - This will display a screen showing the moves that have happened on the past for the person signed on.
- **HISTALL** - Display history entries for all users. This will only work for staff responsible for moves and selected others.

### MMS ALARS Load Library Name

**CV10** - RMVMV.RMV.LOADLIB - TEST BATCH LOAD
- RMVOL.JMV.MODLIB - TEST ONLINE LOAD
- RMVMV.RMV.HLDMOD - TEMPORARY HOLD BATCH LOAD FOR MOVE

**CV13** - RMVMV.JMV.OFCLIB - SPECIAL BATCH LOAD
- RMVOL.JMV.OFCLIB - SPECIAL ONLINE LOAD

**CV14** - RMVMV.JMV.OFCLIB - MIRROR BATCH LOAD
- RMVOL.JMV.OFCLIB - MIRROR ONLINE LOAD

**CV12** - RMVMV.JMV.USRLIB - MODEL BATCH LOAD
- RMVOL.JMV.USRLIB - MODEL ONLINE LOAD

**CV11** - RMVMV.JMV.USRLIB - PRODUCTION BATCH LOAD
- RMVOL.JMV.USRLIB - PRODUCTION ONLINE LOAD


**UMS Load Library Name**

<table>
<thead>
<tr>
<th>Name</th>
<th>Library Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV10</td>
<td>RMVMV.RMV.UMSLOAD</td>
<td>TEST BATCH AND ONLINE LOAD</td>
</tr>
<tr>
<td>CV13</td>
<td>RMVMV.RMV.UMSLOAD2</td>
<td>CONCATENATION SPECIAL BATCH AND ONLINE LOAD</td>
</tr>
<tr>
<td>CV13</td>
<td>RMVMV.RMV.UMSLOAD3</td>
<td>SPECIAL BATCH AND ONLINE LOAD</td>
</tr>
<tr>
<td>CV14</td>
<td></td>
<td>MIRROR BATCH AND ONLINE LOAD</td>
</tr>
<tr>
<td>CV12</td>
<td>RMVMV.RMV.UMSLOADM</td>
<td>MODEL BATCH AND ONLINE LOAD</td>
</tr>
<tr>
<td>CV11</td>
<td>RMVMV.RMV.UMSLOADP</td>
<td>PRODUCTION BATCH AND ONLINE LOAD</td>
</tr>
</tbody>
</table>

**Concatenation Load Libraries for the TEST Region**

<table>
<thead>
<tr>
<th>Library Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMVOL.JMV.MODLIB</td>
<td>ALARS ONLINE TEST LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.TMPLIB</td>
<td>ALARS ONLINE TEMP LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD</td>
<td>UMS ONLINE TEST LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD2</td>
<td>UMS ONLINE SPECIAL LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD3</td>
<td>UMS ONLINE MIRROR LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.USRLIB</td>
<td>PRODUCTION ONLINE LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOADADM</td>
<td>MODEL</td>
</tr>
</tbody>
</table>

**Concatenation Load Libraries for the SPCL Region**

<table>
<thead>
<tr>
<th>Library Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMVOL.JMV.OFCLIB</td>
<td>ALARS ONLINE SPECIAL/MIRROR LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.TMPLIB</td>
<td>ALARS ONLINE TEMP LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD2</td>
<td>UMS SPECIAL LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD3</td>
<td>UMS MIRROR LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.USRLIB</td>
<td>PRODUCTION LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOADADM</td>
<td>UMS MODEL LOAD</td>
</tr>
</tbody>
</table>
Concatenation Load Libraries for the MIRR Region

RMVOL.JMV.OFCLIB - ALARS ONLINE SPECIAL/MIRROR LOAD
RMVOL.JMV.TMPLIB - ALARS ONLINE TEMP LOAD
RMVMV.RMV.UMSLOAD3 - UMS MIRROR LOAD
RMVOL.JMV.USRLIB - PRODUCTION LOAD

Concatenation Load Libraries for the MODL Region

RMVMV.RMV.UMSLOADM - CV12 UMS TEST LOAD
RMVMV.RMV.PRODLOAD - CONSOLIDATED LOAD
RMVOL. RMV.TMPLIB - TEMPORARY LOAD
RMVMV.RMV.UMSLOADP - UMS PRODUCTION

Concatenation Load Libraries for the PROD/ADMIN Region

RMVMV.RMV.PRODLIB - PRODUCTION LOAD

Libraries for Test (CV10) Region

RMVMV.RMV.UMSMACRO - CV10 UMS MACRO
RMVMV.RMV.NDRMACRO - CV10 ALARS MACRO
RMVMV.RMV.UMSSRCE - CV10 UMS SOURCE
RMVMV.RMV.NDRSRCE - CV10 ALARS SORCE
RMVMV.RMV.UMSLOAD - CV10 UMS LOAD
RMVMV.RMV.LOADLIB - CV10 ALARS BATCH LOAD
RMVOL.JMV.MODLIB - CV10 ALARS ONLINE LOAD
RMVMV.RMV.HLDSOR - CV10 HOLD BATCH AND ONLINE SOURCE FOR MMS MOVE
RMVMV.RMV.HLDMOD - CV10 HOLD BATCH AND ONLINE LOAD FOR MMS MOVE
RMVMV.RMV.UMSMACRO.BKUP - BACKUP CV10 UMS MACRO
RMVMV.RMV.UMSSRCE.BKUP - BACKUP CV10 UMS SOURCE
RMVMV.RMV.UMSLOAD.BKUP - BACKUP CV10 UMS LOAD
### Libraries for SPCL (CV13) Region

<table>
<thead>
<tr>
<th>Library Path</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMVMV.RMV.UMSMACR2</td>
<td>CV13 UMS MACRO</td>
</tr>
<tr>
<td>N/A</td>
<td>CV13 ALARS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCE2</td>
<td>CV13 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.JMV.CV13SOR</td>
<td>CV13 ALARS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD2</td>
<td>CV13 UMS LOAD</td>
</tr>
<tr>
<td>RMVMV.JMV.OFCLIB</td>
<td>CV13 ALARS BATCH LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.OFCLIB</td>
<td>CV13 ALARS ONLINE LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSMACR2.BKUP</td>
<td>BACKUP CV13 UMS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCE2.BKUP</td>
<td>BACKUP CV13 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD2.BKUP</td>
<td>BACKUP CV13 UMS LOAD</td>
</tr>
</tbody>
</table>

### Libraries for MIRR (CV14) Region

<table>
<thead>
<tr>
<th>Library Path</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMVMV.RMV.UMSMACR3</td>
<td>CV14 UMS MACRO</td>
</tr>
<tr>
<td>N/A</td>
<td>CV14 ALARS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCE3</td>
<td>CV14 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.JMV.CV13SOR</td>
<td>CV14 ALARS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD3</td>
<td>CV14 UMS LOAD</td>
</tr>
<tr>
<td>RMVMV.JMV.OFCLIB</td>
<td>CV14 ALARS BATCH LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.OFCLIB</td>
<td>CV14 ALARS ONLINE LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSMACR3.BKUP</td>
<td>BACKUP CV14 UMS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCE3.BKUP</td>
<td>BACKUP CV14 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOAD3.BKUP</td>
<td>BACKUP CV14 UMS LOAD</td>
</tr>
</tbody>
</table>

### Libraries for ADMIN/PROD (CV11) Regions

<table>
<thead>
<tr>
<th>Library Path</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMVMV.RMV.UMSMACRP</td>
<td>CV11 UMS MACRO</td>
</tr>
<tr>
<td>RMVMV.JMV.MACLIB</td>
<td>CV11 ALARS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCEP</td>
<td>CV11 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.JMV.SORLIB</td>
<td>CV11 ALARS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOADP</td>
<td>CV11 UMS LOAD</td>
</tr>
<tr>
<td>RMVMV.JMV.USRLIB</td>
<td>CV11 ALARS BATCH LOAD</td>
</tr>
<tr>
<td>RMVOL.JMV.USRLIB</td>
<td>CV11 ALARS ONLINE LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.PRODLOAD</td>
<td>CV11 ON-LINE APPLICATION</td>
</tr>
<tr>
<td></td>
<td>LKOADLIB FOR ALL SYSTEMS</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSMACRP.BKUP</td>
<td>BACKUP CV11 UMS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCEP.BKUP</td>
<td>BACKUP CV11 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOADP.BKUP</td>
<td>BACKUP CV11 UMS LOAD</td>
</tr>
</tbody>
</table>
### Libraries for MODL (CV12) Region

<table>
<thead>
<tr>
<th>Library Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMVMV.RMV.UMSLOADM</td>
<td>CV11 UMS LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSMACRP</td>
<td>CV11 UMS MACRO</td>
</tr>
<tr>
<td>RMVMV.JMV.MACLIB</td>
<td>CV11 ALARS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCEP</td>
<td>CV11 UMS SOURCE</td>
</tr>
<tr>
<td>RMVMV.JMV.SORLIB</td>
<td>CV11 ALARS SOURCE</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOADP</td>
<td>CV11 UMS LOAD</td>
</tr>
<tr>
<td>RMVMV.JMV.USRLIB</td>
<td>CV11 ALARS BATCH LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSLOADP.BKUP</td>
<td>BACKUP CV11 UMS LOAD</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSMACRP.BKUP</td>
<td>BACKUP CV11 UMS MACRO</td>
</tr>
<tr>
<td>RMVMV.RMV.UMSSRCEP.BKUP</td>
<td>BACKUP CV11 UMS SOURCE</td>
</tr>
</tbody>
</table>

### Module Movement

**Libraries for each CICS**

- **CV10 (TEST)**
  - RMVMV.RMV.UMSSRCE
  - RMVMV.RMV.UMSLOAD
  - RMVMV.RMV.UMSMACRO

- **CV13 (SPCL)**
  - RMVMV.RMV.UMSSRCE2
  - RMVMV.RMV.UMSLOAD2
  - RMVMV.RMV.UMSMACR2

- **CV14 (MIRR)**
  - RMVMV.RMV.UMSSRCE3
  - RMVMV.RMV.UMSLOAD3
  - RMVMV.RMV.UMSMACR3

- **CV11 (ADMIN) - (PROD)**
  - RMVMV.RMV.UMSSRCEP
  - RMVMV.RMV.UMSLOADP
  - RMVMV.RMV.UMSMACRP

- **CV12 (MODL)**
  - RMVMV.RMV.UMSLOADM

Modules are **MOVED** from CV10 to CV14 and CV13.

Modules are **COPIED** from CV14 to CV11.
One of the objectives of the UMS system design is the simplification of application design, coding, maintenance, and release-control. This is especially important where a part of the software (the Guest system) is distributed to outside users. To this end, the system control structure is maintained outside of the application programs. Applications are not to issue CICS Program Control Macros because these involve program-name literals. If one is to change program names or the flow of control then all affected programs must be changed and recompiled. In UMS, UGZ0001P / UHZ0001P (Guest/Host Control Program) in conjunction with the UGZ0002P / UHZ0002P (Guest/Host Control Table is used to control the execution of the host and guest programs that make up the UMS application online system. During initialization of the application, the control program will move the names of the programs designated in the Control Table to the common area for later use by the application programs. The Guest and Host tables, referred to as PCTEG and PCTEH after the name of their principal macro, vary slightly in structure while serving the same purpose.

Program Control Table Entry Guest (PCTEG):

The PCTEG controls execution of guest side programs and LXTABLE processing. When the user types in a UMS function code and presses enter, the Guest Side Dispatcher (UGZ0001P) will access the PCTEG and, based on the UMS screen function code, pass control to the appropriate guest input program (the first in the program-name list). Following is an example of an entry in the PCTEG for UPOI:

```
PCTEG "UPOI," PGM = (UGU1011P, UGU1012P), XLATE = UGU1010T
```

PCTEG The macro invocation for this entry.

UPOI Is the UMS screen function code for this entry.

PGM= Designates the guest program name-list.

XLATE Names the LX table for the UPOI function.

Warning: Be certain to review the detail discussion that follows, as there are more parameters than are shown above.

The table-driven control structure facilitates versioning of programs and functions. The following example illustrates the steps necessary to update the PCTEG to accommodate versioning of the UPH screen from release one to two.
Source Code Example - PCTEG

This example is the source for the assembled code that follows and its hexadecimal dump.

**Note:** These examples have been slightly modified by text-edit to allow them to fit in the page or to improve their readability. For brevity, some macros have a number of parameters to show the results of expansion. Some combinations, while assembled correctly may be illogical. Please consult the detailed write-up for parameter usage.

```
UGZ0002P CSECT
NUMENT DC F'0' NUMBER OF ENTRIES
  DC A(FIRST) ADDRESS OF FIRST ENTRY
NUMXFER DC F'0' NUMBER OF XFER (DUPKEY) ENTRIES
  DC A(XFER1) ADDRESS OF FIRST XFER ENTRY
NUMALIAS DC F'0' NUMBER OF ALIAS ENTRIES
  DC A(ALIAS1) ADDRESS OF FIRST ALIAS ENTRY
UGZ0002Q CSECT
XFER1 EQU * FIRST XFER ENTRY
UGZ0002R CSECT
ALIAS1 EQU * FIRST ALIAS ENTRY
UGZ0002P CSECT
*
FIRST DS 0F
PCTEG 'ALAR',PGM=UGZ0009P,TASK=ALAR,XFER=YES
*
PCTEG 'ALAS',PGM=(UGZ0009P,XFERTEST),XFER=DATA
*
PCTEG 'LN ',PGM=(UGL0020P,UGL0021P,UGL0022P),
  RESCROLL=YES,DUPKEY=LI04,ALIAS='ULN ',
  XLATE=UGL0020T,UPDATE=YES,
  FLAGS=FF0000000000EE11,CLEAN=CLEANPGM,
  HOSTA=YES,SCROLL=YES
*
PCTEG 'LNO ',PGM=(UGL0240P,UGL0241P),HOSTA=YES,
  RESCROLL=YES,DUPKEY=LD08,XLATE=UGL0240T,
  SCROLL=YES,ALIAS='ULNO'
*
PCTEG 'STAT',PGM=UGZ0010P,HOSTA=NO,
  FLAGS=FF00000000000000
```
Note: Change in Use for UMS

**Part of the original intent** of UGZ0002P was to provide a reference to all application (3rd character of name not ‘Z’) modules required by the guest, and to require all references to these module names to be indirect through UGZ0002P.

A decision was made for the UMS Release 5.0 to send all modules to the Insurance Companies. Part of the reason was based on **problems**, which ensued when some programmers inserted hard coded reference to module names without indirect references through UGZ0002P with the resulting release not including adequate modules.

Therefore, there is no further need to run the program as **part of the UMS Tape Release procedure** to identify programs by the TYPE parameter. However, all other uses for the module are still very much intact with regard to PCTEG/PCTEH, which controls execution of the programs and LXTABLE processing.
The PCTEG in Detail

UGZ0002P is the module also known as PCTEG, from the name of the macro used in it. This module controls the functions that can be entered at the guest site on the screen in the field labeled FUNCTION. This module needs to move up the “ladder” to the higher regions as new functions are moved up for further integrated testing.

When current functions are being replaced by “new” functions, additional table entries are needed to allow both the current and “new” function to co-exist in the same region (This assumes that module names have changed, but function has remained the same). For example, there is a “new” version of LI. The current entry would be used to create an entry called TLI (the entries have to be in ascending alphabetical order). The TLI entry would need to have the alias parameter removed if it existed. The “new” LI entry would look like the old, except for the program name changes and any other parameter changes.

After the table changes are made, the table must be reassembled and linked into the proper test load library.

```
PCTEG <tskcode>,PGM=,TASK=,XLATE=,FLAGS=,CLEAN=, ALIAS=,SCROLL=NO,RESCROLL=NO,UPDATE=NO, HOSTA=NO,XFER=NO,PF4=,PF7AND8=NO, DUPKEY=,INTERNAL=NO,TYPE=ALL
```

**Field Definitions**

- `<tskcode>`: A 4-byte task code within quotes. PCTE entries must be sequenced by this code. Assumed to be a test entry if the first byte is “T.”

- `ALIAS=<null>` | `<altskcode>`: Optional alternate task code for this function. Same rules as name, except sequencing is not required/appropriate. It may not duplicate any NAME parm. If ALIAS is specified, INTERNAL must equal NO.

- `PGM=(<pgmlist>)`: 1 to 3 processing program names separated by commas. 1 name is required. A given application program may be the first program for one non-test entry and one test entry, only.

- `TASK=<optaskcd>`: A 4-byte optional dispatch task (transaction) code (or name) (4 ebcdic or 8 hex digits).

- `XLATE=<null>` | `<lxmodname>`: Name of optional map/demap translation table. 1 to 8 characters long.
FLAGS=<null> | <flags> An 8-byte optional flagset (expressed in hex, 16 digits) (not allowed if F4 or F9 strings present except as surrogate types). These are available to the programs in this function and may be used to direct processing based on the function code that invoked the program(s). The contents are strictly application dependent.

CLEAN=<null> | <clnpgm> Name of optional tail-end cleanup program, 1 to 8 characters. If omitted the system clean-up program UGZ0005P is used. It is normal to omit this parameter.

SCROLL=YES/NO If YES, program accepts F7/F8 and issues F9/F4.

RESCROLL=YES/NO If program has SCROLL=YES and can accept rescroll.

UPDATE=YES/NO Program accepts F12 for update

HOSTA=YES/NO Program requires host interface area

F4=<null> | <sgcldlist> String of 1 to 4 surrogate codes allowed for F4 entry. Not allowed if SCROLL=YES. No quotes, no commas.

F9=<null> | <sgcldlist> String of 1 to 4 surrogate codes allowed for F9 entry. Not allowed if SCROLL=YES. No quotes, no commas.

F7AND8=YES/NO If YES, pgm accepts F7/F8 but is not a scroll pgm.

DUPKEY=<null>| <hostcd> The 4-byte host duplicate-key function associated with this guest function. If used, HOSTA=YES is required.

INTERNL=YES/NO Is this function restricted to internal dispatch (as opposed to user or keyboard initiated dispatch)?

TYPE=ALL | RMV This specifies if this function is applicable to all sites or just RMV. It determines the relationship of this entry to the global assembly options in sysparm. See the following Sysparm Option discussion.

XFER=YES/NO/DATA This directs the immediate transfer of control to a task-code or program external to the UMS Guest system. The PGM parameter must have UGZ0009P as the first program and HOSTA=NO. If transfer is to a task, than TASK=<xfertask>, where <xfertask> is the transid to start, is required. Control passes by a CICS START. If control goes to a program (<xferpgm>) then its name will be the second value in the PGM= parameter; ie. PGM=(UGZ0009p,<xferpgm>). Control will pass by CICS XCTL. YES and DATA are synomonous except that the DATA option passes a 23 byte (comm) data area. When this option is specified, the PGM= option must specify the foreign interface module (UGZ0009P) as
the first program. A value must be specified for the TASK= option, and a value may be specified for a second program name. If XFER=DATA, a parameter-area of the following format is built by the UMS control-dispatch mechanism for the foreign application:

<table>
<thead>
<tr>
<th>bytes</th>
<th>len</th>
<th>usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-03</td>
<td>4</td>
<td>ID used to sign-on to UMS</td>
</tr>
<tr>
<td>04-11</td>
<td>8</td>
<td>Password used to sign-on to UMS</td>
</tr>
<tr>
<td>12-15</td>
<td>4</td>
<td>UMS function-code entered</td>
</tr>
<tr>
<td>16-17</td>
<td>2</td>
<td>UMS entry-code</td>
</tr>
<tr>
<td>18-22</td>
<td>5</td>
<td>UMS surrogate-type and code if F4 or F9</td>
</tr>
</tbody>
</table>

total length: 23 bytes

If control-dispatch detects that no second program-name was specified, it starts a terminal attached transaction (taking the name from the TASK= operand) in behalf of the terminal. If XFER=DATA was specified, the 23-byte parameter-area is passed as start-data, otherwise there is none. If control-dispatch detects that a second program-name was specified, it XCTL's to the specified program. Only if XFER=DATA is specified, is the 23-byte parameter-area is passed as a common-area, otherwise no common-area is passed. The invoked transaction runs under the transaction-id UMS was running under.

The Guest common area is freed before the transfer.

**SYSPARM Options**

The use of the SYSPARM option allows the creation of tailored output from the assembly of the Program Control Table. The output can be:

1. Full load module with all entries.
2. Load module with RMV-peculiar entries.
3. Generate delete control cards only.

One enters the sysparm into the standard assembly proc as follows:

\[
CPARM2=',SYSPARM=<umstblopt>\]'

where \(<umstblopt>\) may be as follows:

\(<umstblopt>=\text{GENERATE\_ALL}\)
The full PCTEG will be generated. This is the default if no sysparm is present.

<umstblopt>=GENERATE_NOT_RMV

A partial PCTEG will be generated. All entries with TYPE=RMV will be excluded from the generation.

<umstblopt>=PUNCH_DELETE_DSN=<data-set-name>

No PCTEG will be generated. Card images of the form:

DELETE <data-set-name>(<member-name>)

will be generated for all modules referenced on the PCTEG/TYPE= statement. These cards may be used as IDCAMS input. This option will also raise CONDITION-CODE=7, largely to inhibit execution of the LKED step in the standard assembly proc.

Example:

SYSPARM=PUNCH_DELETE_DSN=RMVMV.UMS.LOADLIB

might generate, among others:

DELETE RMVMV.UMS.LOADLIB(UGZ0002P)
Assembled Code Example - PCTEG

Note: These examples have been slightly modified by text-edit to allow them to fit in the page or to improve their readability. For brevity, some macros have a number of parameters to show the results of expansion. Some combinations, while assembled correctly may be illogical. Please consult the detailed write-up for parameter usage.

EXTERNAL SYMBOL DICTIONARY

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>ID</th>
<th>ADDR</th>
<th>LENGTH</th>
<th>LD ID</th>
<th>FLAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UGZ0002P</td>
<td>SD</td>
<td>0001</td>
<td>000000</td>
<td>000215</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>UGZ0002Q</td>
<td>SD</td>
<td>0002</td>
<td>000218</td>
<td>000018</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>UGZ0002R</td>
<td>SD</td>
<td>0003</td>
<td>000230</td>
<td>000010</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

000000 00000000
000004 00000018
000008 00000000
00000C 00000218
000010 00000000
000014 00000230
000218 00218
000230 00230
000018 000018

000218 000218
000230 000230
000018 000018

00218 00218
00230 00230
00018 00018

03810000
03820000
03830000
03840000
03850000
03860000
03870000
03880000
03890000
03900000
03910000
03920000
03930000
03940000
03950000
Chapter Five - UMS Program Control Tables


000018 386+SYM1 DS OCL1 01-00282
000018 C1D3C1D9C1D3C1D9 387+ DC CL4'ALAR',CL4'ALAR' 01-00292
000020 E4C7E9F0F0F0F0D7 388+GST0001 DC CL8'UGZ0009P',CL8' 01-00293
000030 4040404040404040 DC CL8',CL8' 01-00294
000048 0000000000000000 DC AL1(0),AL1(0),AL2(0) 01-00295
00004C 0000000000000000 DC XL8'0000000000000000' 01-00300
000054 394 * 03960001
000054 C1D3C1E240404040 DC CL4'ALAS',CL4' 01-00292
000058 40000000 DC CL8',CL8' 01-00294
00006C 4040404040404040 DC AL1(32),AL1(0),AL2(0) 01-00295
00007C 4040404040404040 DC XL8'0000000000000000' 01-00300
000084 20000000 DC AL1(0),AL1(0),AL2(0) 01-00295
000088 0000000000000000 DC XL8'0000000000000000' 01-00300
000094 20000000 DC AL1(32),AL1(0),AL2(0) 01-00295
000098 0000000000000000 DC XL8'0000000000000000' 01-00300
00009C E4C7D3F0F0F0F0D7 411+UGL0020P DC CL8'UGL0020P',CL8'UGL0021P' 01-00293
00009E E4C7D3F0F0F0F0D7 412+ DC CL8'UGL0022P',CL8'CLEANPGM' 01-00294

400 * 03990001
404 * 04000001
PCTEG 'LN ',ALIAS='ULN ', PGM=(UGL0020P,UGL0021P,UGL0022P), X
XFER=DATA 03980001
RESCALL=YES,DUPKEY=LI04, X
XLATE=UGL0020T,UPDATE=YES, X04030003
FLAGS=FF0000000000EE11, X
CLEAN=CLEANPGM, X04040002
HOSTA=YES,SCROLL=YES 04050002

000218 406+UGZ0002Q CSECT 01-00267
000218 D3C9F0F0400000090 407+ DC CL4'LI04',AL4(SYS$LI04) 01-00268
000090 408+UGZ0002P CSECT 01-00269
000090 409+AL10015 EQU * 01-00290
000090 D3D5404040404040 410+SYS$LI04 DC CL4'LN ',CL4' 01-00292
000098 E4C7D3F0F0F2F0F7 411+UGL0020P DC CL8'UGL0020P',CL8'UGL0021P' 01-00293
0000A8 E4C7D3F0F0F2F0F7 412+ DC CL8'UGL0022P',CL8'CLEANPGM' 01-00294

Chapter Five - UMS Program Control Tables

63

0000B8 E4C7D3F0F0F2F0E3 413+ DC CL8'UGL0020T' 01-00295
0000C0 C70000000 414+ DC AL1(199),AL1(0),AL2(0) 01-00296
0000C4 FF0000000000EE11 415+ DC XL8'FF0000000000EE11' 01-00301

416 *
417 *

418 PCTEG 'LNO ',PGM=(UGL0240P,UGL0241P), HOSTA=YES,XLATE=UGL0240T, RESCROLL=YES,DUPKEY=LD08, SCROLL=YES,ALIAS='ULNO'

000220 419+UGZ0002Q CSECT 01-00267

000220 D3C4F0F80000000CC 420+ DC CL4'LD08',AL4(SYM$LD08) 01-00268
0000CC 421+UGZ0002P CSECT 01-00269

000CC 422+ALI0022 EQU * 01-00290

0000CC D3D5D64040404040 423+SYM$LD08 DC CL4'LNO ',CL4'

0000D4 E4C7D3F0F2F0F0D7 424+PUGL0240P DC CL8'UGL0240P',CL8'UGL0241P' 01-00293
0000E4 425+ DC CL8' ',CL8' ' 01-00294

0000F4 E4C7D3F0F2F0F0E3 426+ DC CL8'UGL0240T' 01-00295
0000FC C30000000 427+ DC AL1(195),AL1(0),AL2(0) 01-00296
000100 0000000000000000 428+ DC XL8'0000000000000000' 01-00301

000108 E2E3C1E340404040 429 *

000110 E4C7E9F0F0F1F0D7 430 *

000120 4040404040404040 431 PCTEG 'STAT',PGM=UGZ0010P,HOSTA=NO, FLAGS=FF000000000000 01-00292

000130 4040404040404040 432+ DC CL4'STAT',CL4' ' 01-00293

000138 0000000000 433+GST0029 DC CL8'UGZ0010P',CL8' ' 01-00294

00013C FF00000000000000 434+ DC CL8' ',CL8' ' 01-00295

435+ DC CL8' '

436+ DC AL1(0),AL1(0),AL2(0) 01-00296

437+ DC XL8'FF00000000000000' 01-00301

438 *
439 *

440 PCTEG 'TPIC',PGM=(UGUI120P,UGUO120P), RESCROLL=YES,DUPKEY=LD18, X

XLATE=UGUX120T,HOSTA=YES,PF9=01, X

Chapter Five - UMS Program Control Tables 63
SCROLL=YES, INTERNL=YES, UPDATE=YES 04190004

000228 441+UGZ0002Q CSECT
000228 D3C4F1F800000144 01-00267
000228 442+ DC CL4'LD18', AL4(SYM$LD18) 01-00268
000144 443+UGZ0002P CSECT
000144 444+SYM$LD18 DC CL4'TPIC', CL4' ' 01-00269
000144 445+TUGUI120P DC CL8'UGI120P', CL8'UGO120P' 01-00293
00015C 446+ DC CL8', ', CL8' ' 01-00294
00016C 447+ DC CL8'UGUX120T' 01-00295
000174 448+ DC AL1(215), AL1(1), AL2(0) 01-00296
000178 449+ DC CL4',', CL4'01' 01-00298
000180 E3D9C94040404040 01-00299
000188 E4C7D9C9F1F1F0D7 01-00292
000198 450+ DC CL4'TRI', CL4' ' 01-00292
0001A8 E4C7D9E7F1F1F0E3 01-00293
0001B0 451+ DC CL4'TRI', CL4' ' 01-00292
0001B4 F1F0F2F4F1F0F2F4 01-00293
0001BC E5C5D94040404040 01-00293
0001C4 E4C7E9F0F0F3F2D7 01-00294
0001D4 462+ DC CL4'VER', CL4' ' 01-00292
0001E4 463+SYM$LD18 DC CL4'TPIC', CL4' ' 01-00292
0001EC 464+ DC CL8'UGI120P', CL8'UGO120P' 01-00293
0001F0 465+ DC CL8', ', CL8' ' 01-00294
0001F8 FFFFFFFF 01-00294
000230 470 LAST DC XL4'FFFFFFFF' 01-00301
000230 471 UGZ0002Q CSECT 01-00302
Chapter Five - UMS Program Control Tables

RELOCATION DICTIONARY

<table>
<thead>
<tr>
<th>POS.ID</th>
<th>REL.ID</th>
<th>FLAGS</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0001</td>
<td>0C</td>
<td>000004</td>
</tr>
<tr>
<td>0001</td>
<td>0002</td>
<td>0C</td>
<td>00000C</td>
</tr>
<tr>
<td>0001</td>
<td>0003</td>
<td>0C</td>
<td>000014</td>
</tr>
<tr>
<td>0002</td>
<td>0001</td>
<td>0C</td>
<td>00021C</td>
</tr>
<tr>
<td>0002</td>
<td>0001</td>
<td>0C</td>
<td>000224</td>
</tr>
<tr>
<td>0003</td>
<td>0001</td>
<td>0C</td>
<td>000234</td>
</tr>
<tr>
<td>0003</td>
<td>0001</td>
<td>0C</td>
<td>00023C</td>
</tr>
</tbody>
</table>

RELOCATION DICTIONARY

<table>
<thead>
<tr>
<th>POS.ID</th>
<th>REL.ID</th>
<th>FLAGS</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>00230</td>
<td>472</td>
<td>XFER2</td>
<td>EQU</td>
</tr>
<tr>
<td>0001FC</td>
<td>473</td>
<td>UGZ0002P</td>
<td>CSECT</td>
</tr>
<tr>
<td>0001FC</td>
<td>474</td>
<td>ORG</td>
<td>NUMENT</td>
</tr>
<tr>
<td>0000000000000008</td>
<td>475</td>
<td>DC</td>
<td>A((LAST-FIRST)/(SYM2-SYM1))</td>
</tr>
<tr>
<td>000004</td>
<td>476</td>
<td>ORG</td>
<td>NUMXFER</td>
</tr>
<tr>
<td>000008</td>
<td>00000003</td>
<td>477</td>
<td>DC</td>
</tr>
<tr>
<td>000230</td>
<td>478</td>
<td>CLEANUP</td>
<td></td>
</tr>
<tr>
<td>00230</td>
<td>479+UGZ0002R</td>
<td>CSECT</td>
<td></td>
</tr>
<tr>
<td>000230</td>
<td>E4D3D54000000090</td>
<td>480+</td>
<td>DC</td>
</tr>
<tr>
<td>000238</td>
<td>E4D3D5D600000000CC</td>
<td>481+</td>
<td>DC</td>
</tr>
<tr>
<td>00240</td>
<td>482+ALIEND</td>
<td>EQU</td>
<td>*</td>
</tr>
<tr>
<td>00000C</td>
<td>483+UGZ0002P</td>
<td>CSECT</td>
<td></td>
</tr>
<tr>
<td>00000C</td>
<td>00010</td>
<td>484+</td>
<td>ORG</td>
</tr>
<tr>
<td>00002</td>
<td>485+ALICOUNT</td>
<td>EQU</td>
<td>(ALIEND-ALIAS1)/8</td>
</tr>
<tr>
<td>000010 00000002</td>
<td>486+</td>
<td>DC</td>
<td>AL4(ALICOUNT)</td>
</tr>
<tr>
<td>000014</td>
<td>001FC</td>
<td>487+</td>
<td>ORG</td>
</tr>
<tr>
<td>0001FC</td>
<td>488</td>
<td>ORG</td>
<td></td>
</tr>
<tr>
<td>0001FC</td>
<td>489</td>
<td>DC</td>
<td>CL8'&amp;SYSDATE',CL1' ',CL5'&amp;SYSTIME'</td>
</tr>
<tr>
<td>0001FC</td>
<td>F0F761F2F361F9F1</td>
<td>+</td>
<td>DC CL8'07/23/91',CL1' ',CL5'11.28'</td>
</tr>
<tr>
<td>00020A</td>
<td>7EE5C5D940F0F34B</td>
<td>490</td>
<td>DC</td>
</tr>
<tr>
<td>00020A</td>
<td>491</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

ASSEMBLER SUMMARY

<table>
<thead>
<tr>
<th>POS.ID</th>
<th>REL.ID</th>
<th>FLAGS</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0001</td>
<td>0C</td>
<td>000004</td>
</tr>
<tr>
<td>0001</td>
<td>0002</td>
<td>0C</td>
<td>00000C</td>
</tr>
<tr>
<td>0001</td>
<td>0003</td>
<td>0C</td>
<td>000014</td>
</tr>
<tr>
<td>0002</td>
<td>0001</td>
<td>0C</td>
<td>00021C</td>
</tr>
<tr>
<td>0002</td>
<td>0001</td>
<td>0C</td>
<td>000224</td>
</tr>
<tr>
<td>0003</td>
<td>0001</td>
<td>0C</td>
<td>000234</td>
</tr>
<tr>
<td>0003</td>
<td>0001</td>
<td>0C</td>
<td>00023C</td>
</tr>
</tbody>
</table>

DIAGNOSTIC CROSS REFERENCE AND ASSEMBLER SUMMA
NO STATEMENTS FLAGGED IN THIS ASSEMBLY

OVERRIDING PARAMETERS – LIST, NODECK, LOAD, LINECOUNT(65)

OPTIONS FOR THIS ASSEMBLY
NODECK, OBJECT, LIST, XREF(FULL), NORENT, NOTEST, NOBATCH, ALIGN, ESD, RLD, NOTERM, NODBCS,
LINECOUNT(65), FLAG(0), SYSPARM()

NO OVERRIDING DD NAMES
433 CARDS FROM SYSIN       0 CARDS FROM SYSLIB
584 LINES OUTPUT          23 CARDS OUTPUT

JOB RMCJEBBE  STEP STEP010  PROCEDURE LKED
INVOCATION PARAMETERS – LIST, XREF
ACTUAL SIZE=(317440,79872)
OUTPUT DATA SET IS ON VOLUME RELP02

CROSS REFERENCE TABLE

<table>
<thead>
<tr>
<th>CONTROL SECTION</th>
<th>ENTRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>ORIGIN</td>
</tr>
<tr>
<td>UGZ0002P</td>
<td>00</td>
</tr>
<tr>
<td>UGZ0002Q</td>
<td>218</td>
</tr>
<tr>
<td>UGZ0002R</td>
<td>230</td>
</tr>
</tbody>
</table>

LOCATION REFERS TO SYMBOL IN CONTROL SECTION
LOCATION REFERS TO SYMBOL IN CONTROL
SECTION

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>REFERS TO</th>
<th>SYMBOL IN CONTROL SECTION</th>
<th>LOCATION</th>
<th>REFERS TO</th>
<th>SYMBOL IN CONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>UGZ0002Q</td>
<td>UGZ0002Q</td>
<td>14</td>
<td>UGZ0002R</td>
<td>UGZ0002R</td>
</tr>
<tr>
<td>21C</td>
<td>UGZ0002P</td>
<td>UGZ0002P</td>
<td>224</td>
<td>UGZ0002P</td>
<td>UGZ0002P</td>
</tr>
<tr>
<td>22C</td>
<td>UGZ0002P</td>
<td>UGZ0002P</td>
<td>234</td>
<td>UGZ0002P</td>
<td>UGZ0002P</td>
</tr>
<tr>
<td>23C</td>
<td>UGZ0002P</td>
<td>UGZ0002P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ENTRY ADDRESS 00
TOTAL LENGTH 240
** TST0002P REPLACED AND HAS AMODE 24
** LOAD MODULE HAS RMODE 24
** AUTHORIZATION CODE IS 0.
Hexadecimal Dump of PCTEG Example

AMASPZAP INSPECTS, MODIFIES, AND DUMPS CSECTS OR SPECIFIC DATA RECORDS ON DIRECT ACCESS STORAGE.
DUMPT TST0002P ALL 00110005

**CCHHR- 0011000419 RECORD LENGTH- 000240 MEMBER NAME TST0002P CSECT NAME UGZ0002P
000000 00000008 00000018 00000003 00000218 00000002 00000230 C1D3C1D9 C1D3C1D9 *...............*
000020 E4C7E9F0 F0F0F9D7 40404040 40404040 40404040 40404040 40404040 *UGZ0009P* *
000040 40404040 40404040 00000000 00000000 00000000 00000000 C1D3C1E2 40404040 E4C7E9F0 * *
000060 F0F0F9D7 E7C6C5D9 E3C5E2E3 40404040 40404040 40404040 40404040 *009PXFERTTEST*
000080 40404040 20000000 00000000 00000000 00000000 D3D54040 40404040 E4C7D3F0 F0F2F0D7 *
0000A0 E4C7D3F0 F0F2F1D7 E4C7D3F0 F0F2F2D7 C3D3C5C1 D5D7C7D4 E4C7D3F0 F0F2F0E3 *
0000C0 C7000000 FF000000 0000EE11 D3D5D640 40404040 E4C7D3F0 F2F4F0D7 E4C7D3F0 *G...........LNO* *
0000E0 F2F4F1D7 40404040 40404040 40404040 40404040 40404040 40404040 E4C7D3F0 F2F4F0E3 C3000000 *241P *
000100 00000000 00000000 E2E3C1E3 40404040 E4C7E9F0 F0F1F0D7 40404040 40404040 *..........STAT* *
000120 40404040 40404040 40404040 40404040 40404040 40404040 40404040 FF000000 *
000140 00000000 E3D7C9C3 40404040 E4C7E4C9 F1F2F0D7 E4C7E4D6 F1F2F0D7 40404040 *
000160 40404040 40404040 40404040 E4C7E4E7 F1F2F0E3 D7010000 40404040 F0F14040 *
000180 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
0001A0 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *
0001C0 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
0001E0 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
000200 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
000220 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
000240 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
000260 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
000280 00000000 00000000 E4C7E9F0 40404040 40404040 40404040 40404040 40404040 F0F14040 *....TPIC  UGUI*
AMA113I COMPLETED DUMP REQUIREMENTS

AMA100I AMASPZAP PROCESSING COMPLETED

****************************************************************************** BOTTOM OF DATA******************************************************************************
Program Control Table Entry Host (PCTEH)

The PCTEH controls execution of the host side programs, service modules, and duplicate resolution modules. When control is passed from the guest side to the host side, C&C’s Host Side Dispatcher (UHZ0001P) accesses the PCTEH. The internal function code (refer to Internal Function Codes section for more detail) passes control to the appropriate host input program. UHZ0001P will move the names of all host side programs, service modules, and duplicate resolution modules to the common area for later use by the guest side programs. If the Host processing receives anything other than a surrogate key, the first program in the <name-list> will try to convert it into a unique surrogate key. If none is found, an error is returned. If only one is found, processing will pass to the second program in the <name-list>. If 2 or more are found, the duplicate resolver transaction will be invoked to provide scroll data and return to the Guest.

The SERVICE macro defines groups of service program names so that a simple reference in the PCTEH macro will allow the task being defined access to that set of service modules. The form is:

```
SERVICE <name>,PGM=(<name-list>)
```

- `<name>` a 1 or 2 digit identifier for this service set.
- `<name-list>` 1 to 8 service program names. 1 required. Need not be the first.

The PCTEH macro has a similar function to the PCTEG macro on the Guest side. That is, it defines the task codes that may be used, the programs that may be invoked for each one along with some additional information. They are assembled and linked as a table (load module) that is accessed by UHZ0001P, the HOST Control/Dispatch Program to manage the system. The form of the macro is as follows:

```
PCTEH <tskcode>,PGM=,TASK=,RMV=,FLAGS=,
       CLEAN=,DUPKEY=,SERVICE=
```

Field Definitions

- `<tskcode>` A 4-byte block type or task code. PCTEG entries must be sequenced by this code. Only alpha-numeric values may be used.
- `PGM=<pgmlist>` 1 to 3 processing program names. 1 REQUIRED.
TASK=<null> | <optaskcd>  A 4-byte optional dispatch task (transaction) code.

RMV=<alfuncn>  An ALARS equivalent function-name (for security mapping)

FLAGS=<null> | <flags>  An 8-byte optional flagset (expressed in hex) These are available to the application programs for this function. They may be used to direct processing based on the function code invoking the program(s).

CLEAN=<null> | <clnpgm>  The Name of an optional tail-end cleanup program.

DUPKEY=<null> | <hostcd>  A 4-byte optional host block-type or task code used for resolution of duplicate keys detected during this invocation. **Note** that specification of a non-existent entry will result in undefined symbols of the form BTC$XXXX where XXXX is the block type or transaction code.

SERVICE=<null> | <srvcset>  The optional 1 or 2 digit identifier of a defined service set, established by a SERVICE macro, used by this function.

Following is an example of an entry in the PCTEH for UPOI:

```
PCTEH  UU11,PGM=(UHU3013P,UHU3014P),RMV=POI,   X
          DUPKEY=LD08,SERVICE=5
```

PCTEH Indicates the beginning of a table entry.

UU11 Used by UHZ0001P to determine which table entry to reference based on the internal function code passed by the guest input program. 'UU11' is one of the internal function codes used in the guest input program for UPOI.

PGM Refers to the host input and output program names.

RMV Used for mapping security features from the ALARS equivalent screen (POI) to UPOI.

DUPKEY Refers to duplicate resolution modules. In this case, 'LD08' references the license duplicate resolution module.

SERVICE Used to indicate that the service modules UHU101CP, UHU1SDIP, and UHL0055P are available to the UPOI host programs.
Control tables allow the UMS online system flexibility to maintain two versions of the application at any given time (refer to the Versioning and Naming Standards section for more detail). Each time versioning is applied to an online UMS screen, it will be necessary to update the PCTEG and PCTEH.

When current functions are being replaced by “new” functions or “new” functions are just being added, the “new” functions are entered to allow both the current and “new” function to co-exist in the same region. For example there is a “new” version of Pers look-up by Mass. license number. The current entry (LI07) would remain untouched. The “new” entry would be called LI20 (Remember, the entries have to be in ascending alphabetical order). The “new” LI20 entry would look like the old except for the program name changes and any other parameter changes.
Source Code Example - PCTEH

UHZ0002P  CSECT
NUMENT  DC  F'0'  NUMBER OF ENTRIES
DC  A(FIRST)  POINT TO 1ST ENTRY
SERVICE 1,PGM=(UHRSTTLS,UHRSREGS)
SERVICE 2,PGM=(UHUC101P,UHUSSDIP,UHL0050P)
SERVICE 3,PGM=(UHRSREGS,UHL0050P)
* NDR/ESI ROUTING TABLE:
SERVICE 4,PGM=(UHL0200P)
SERVICE 5,PGM=(UHU101CP,MMRSSDIP,UHL0055P)  KK0191
SERVICE 6,PGM=(UHL0055P,UHL0065P)  SC0890
SPACE 3
* TABLE LOAD FUNCTION:
PCTEH HTB1,PGM=(UHZ0004P)
PCTEH HTB2,PGM=(UHZ0004P,MRMXWAFL)
SPACE 3
* GENERALIZED FETCHER OF BREFS:
PCTEH LB10,PGM=(UHL0420P),RMV=LI,CLEAN=TSTCLEAN
SPACE 3
* DUPLICATE SSN SCROLL
PCTEH LD04,PGM=(UHL0500P),RMV=LP
SPACE 3
* LICENSE HISTORY LOOK-UP BY LICENSE #
PCTEH LH10,PGM=(UHL0070P,UHL0060P), RMV=LI,DUPKEY=LI04
SPACE 3
* LICENSE NUMBER SCROLL
  PCTEH LI04,PGM=(UHL0020P), RMV=LP
  SPACE 3
  * LICENSE INQUIRY (BY LIC. NUMBER) (NEW)
  PCTEH LI07,PGM=(UHL0070P,UHL0100P,UHL0110P),SERVICE=6,
              RMV=LI,DUPKEY=LI04,FLAGS=0003000000000000,
              CLEAN=TSTCLEAN
  SPACE 3
* LXGHOST INQUIRE_NDR
  PCTEH LX51,PGM=(UHL0300P,UHLDNR1P,UHL0400P),SERVICE=4,RMV=LX,
              FLAGS=8000000000000000
  SPACE 3
* LX-WRITE OVERRIDE BREF: (1ST 2 FLAG BYTES=SUBFUNCTION SECURITY)
  PCTEH LX70,PGM=(UHL0260P,UHL0400P),SERVICE=4,RMV=LX,
              FLAGS=C4400000000000000
  SPACE 3
LAST DC XL4'FFFFFFFF'
ORG NUMENT
DC A((LAST-FIRST)/(SYM2-SYM1))
ORG
DC CL8'&SYSDATE',CL1' ',CL5'&SYSTIME'
END

******************************************************************************

END
Assembled Code Example - PCTEH

Note: These examples have been slightly modified by text-edit to allow them to fit in the page or to improve their readability. For brevity, some macros have a number of parameters to show the results of expansion. Some combinations, while assembled correctly may be illogical. Please consult the detailed write-up for parameter usage.

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM</th>
<th>H</th>
<th>V</th>
<th>02</th>
<th>15.16</th>
<th>07/15/91</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>260</td>
<td>UHZ0002P</td>
<td>CSECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000</td>
<td>000000000</td>
<td>00004</td>
<td>0000188</td>
<td>265</td>
<td>DC</td>
<td>A(FIRST)</td>
<td>POINT TO 1ST ENTRY</td>
<td>02630000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000008 E4C8D9E2E3E3D3E2</td>
<td>264</td>
<td>SRV#1</td>
<td>DC</td>
<td>CL8'UHRSTTLS',CL8'UHRSREGS'</td>
<td>01-00254</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000018 4040404040404040</td>
<td>265+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00255</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000028 4040404040404040</td>
<td>266+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000038 4040404040404040</td>
<td>267+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00257</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000048 E4C8E4C3F1F0F1D7</td>
<td>269</td>
<td>SRV#2</td>
<td>DC</td>
<td>CL8'UHUC101P',CL8'UHUSSDIP'</td>
<td>01-00254</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000058 E4C8D3F0F0F5F0D7</td>
<td>270+</td>
<td>DC</td>
<td>CL8'UHL0050P',CL8'</td>
<td>'</td>
<td>01-00255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000068 4040404040404040</td>
<td>271+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000078 4040404040404040</td>
<td>272+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00257</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000088 E4C8D9E2D9C5C7E2</td>
<td>274</td>
<td>SRV#3</td>
<td>DC</td>
<td>CL8'UHRSREGS',CL8'UHL0050P'</td>
<td>01-00254</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000098 4040404040404040</td>
<td>275+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00255</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000A8 4040404040404040</td>
<td>276+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000B8 4040404040404040</td>
<td>277+</td>
<td>DC</td>
<td>CL8'</td>
<td>'</td>
<td>CL8</td>
<td>'</td>
<td>01-00257</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NDR/ESI ROUTING TABLE:
279     SERVICE 4,PGM=(UHL0200P)                      02680000
0000C8  E4C8D3F0F2F0F0D7
0000D8  4040404040404040
0000E8  4040404040404040
0000F8  4040404040404040
280+SRV#4   DC  CL8'UHL0200P',CL8'    '  01-00254
281+    DC  CL8'    ',CL8'    '  01-00255
282+    DC  CL8'    ',CL8'    '  01-00256
283+    DC  CL8'    ',CL8'    '  01-00257
284     SERVICE 5,PGM=(UHU101CP,MMRSSDIP,UHL0055P)  690005
000108  E4C8E4F1F0F1C3D7
000118  E4C8D3F0F0F5F5D7
000128  4040404040404040
000138  4040404040404040
285+SRV#5   DC  CL8'UHU101CP',CL8'MMRSSDIP'  01-00254
286+    DC  CL8'UHL0055P',CL8'    '  01-00255
287+    DC  CL8'    ',CL8'    '  01-00256
288+    DC  CL8'    ',CL8'    '  01-00257
289     SERVICE 6,PGM=(UHL0055P,UHL0065P)          691000
000148  E4C8D3F0F0F5F5D7
000158  4040404040404040
000168  4040404040404040
000178  4040404040404040
000188  4040404040404040
290+SRV#6   DC  CL8'UHL0055P',CL8'UHL0065P'  01-00254
291+    DC  CL8'    ',CL8'    '  01-00255
292+    DC  CL8'    ',CL8'    '  01-00256
293+    DC  CL8'    ',CL8'    '  01-00257
295 *  02692000
296 * TABLE LOAD FUNCTION:                       02692106
297     PCTEH HTB1,PGM=(UHZ0004P)                 02692300
298+FIRST DS 0F ORIGIN OF FIRST ENTRY 02692400
000188  IEV254 *** MNOTE ***
000188  IEV254 *** MNOTE ***
000188  300+SYM1    DS 0CL1  01-00157
000190  301+BTC$HTB1    DC  CL4'HTB1',CL4'    '  01-00159
00019E  302+    DC  CL6'    ',XL8'000000000000000'  01-00160
0001AE  303+    DC  CL8'UHZ0004P',CL8'    '  01-00161
0001BE  304+    DC  CL8'    ',CL8'    '  01-00162
0001C6  305+    DC  XL4'0',AL4(0)  01-00163
0001C6  306+SYM2    DS 0CL1  01-00165
0001C6  307     PCTEH HTB2,PGM=(UHZ0004P,MRMXWAF1)  02692400
0001C6  IEV254 *** MNOTE ***
0001C6  IEV254 *** MNOTE ***
0001C6  308+    4,WARNING - THIS FUNCTION WILL BE UNSECURED 01-00052
0001C6  IEV254 *** MNOTE ***
0001C6  IEV254 *** MNOTE ***
0001C6  309+BTC$HTB2    DC  CL4'HTB2',CL4'    '  01-00159
0001CE 4040404040400000  310+  DC  CL6'      ',XL8'0000000000000000'  01-00160
0001DC E4C8E9F0F0F04D7  311+  DC  CL8'UHZ0004P',CL8'MRMXWAFL'    01-00161
0001EC 4040404040404040  312+  DC  CL8'      ',CL8'        '    01-00162
0001FC 0000000000000000  313+  DC  XL4'0',AL4(0)                      01-00163

315 * GENERALIZED FETCHER OF BREFS:                  02692800
316           PCTEH LB10,PGM=(UHL0420P),RMV=LI,CLEAN=TSTCLEAN
000204 D3C2F1F040404040  317+BTC$LB10  DC   CL4'LB10',CL4'    '                01-00159
00020C D3D740404000000   318+         DC    CL6'LI',XL8'0000000000000000'      01-00160
00021A E4C8D3F0F4F2F0D7   319+         DC    CL8'UHL0420P',CL8'        '    01-00161
00022A 4040404040404040   320+         DC    CL8'      ',CL8'TSTCLEAN'        01-00162

00023A 0000000000000000                                      PAGE 7
000242 D3C4F0F440404040  321+         DC    XL4'0',AL4(0)                      01-00163

00024A 0000000000000000                                      PAGE 7
000258 E4C8D3F0F5F0F0D7  322 * DUPLICATE SSN SCROLL                   02693100
000268 4040404040404040  323        PCTEH LD04,PGM=(UHL0500P),RMV=LP    02694000
000278 0000000000000000                                      PAGE 7
000280 D3C8F1F040404040  324        PCTEH LH10,PGM=(UHL0070P,UHL0060P),RMV=LI,DUPKEY=LI04 02698000
000288 D3C9404040400000  325        PCTEH LB04,PGM=(UHL0500P),RMV=LP    02694000
000296 E4C8D3F0F0F7F0D7  326+        DC    CL6'LI',XL8'0000000000000000'      01-00160
00029E 4040404040404040  327+        DC    CL8'UHL0070P',CL8'UHL0060P'    01-00161

331 * LICENSE HISTORY LOOK-UP BY LICENSE #             02697000
332        PCTEH LH10,PGM=(UHL0070P,UHL0060P),RMV=LI,DUPKEY=LI04 02698000
002BE 333+CHK$LI040011  EQU BTC$LI04 "UNDEFINED" ERROR IF UNKN TYPE 02699000
000280 D3C8F1F040404040  334+BTC$LB10  DC   CL4'LH10',CL4'        '    01-00159
000288 D3C9404040400000  335+        DC    CL6'LI',XL8'0000000000000000'      01-00160
000296 E4C8D3F0F0F7F0D7  336+        DC    CL8'UHL0070P',CL8'UHL0060P'    01-00161

Chapter Five - UMS Program Control Tables
Chapter Five - UMS Program Control Tables

337+  DC  CL8' ,CL8'        ' 01-00162
338+  DC  CL4'LI04',AL4(0) 01-00163

340 * LICENSE NUMBER SCROLL

341  PCTEH LI04,PGM=(UHL0020P),
     RMV=LP
     01-00158

342+BTC$LI04  DC  CL4'LI04',CL4'        ' 01-00159
343+  DC  CL6'LP',XL8'0000000000000000' 01-00160
344+  DC  CL8'UHL0020P',CL8'        ' 01-00161
345+  DC  CL8'        ',CL8'        ' 01-00162
346+  DC  XL4'0',AL4(0) 01-00163

348 * LICENSE INQUIRY (BY LIC. NUMBER) (NEW)

349  PCTEH LI07,PGM=(UHL0070P,UHL0100P,UHL0110P),
     SERVICE=6,RMV=LI,DUPKEY=LI04,
     CLEAN=TSTCLEAN 02881006

002BE 350+CHK$LI040013  EQU BTC$LI04 "UNDEFINED" ERROR IF UNKN TYPE

351+BTC$LI07  DC  CL4'LI07',CL4'        ' 01-00159
352+  DC  CL6'LI',XL8'0003000000000000' 01-00160
353+  DC  CL8'UHL0070P',CL8'UHL0100P' 01-00161
354+  DC  CL8'UHL0110P',CL8'TSTCLEAN' 01-00162
355+  DC  CL4'LI04',AL4(SRV#6) 01-00163

357 * LXGHOST INQUIRE_NDR

358  PCTEH LX51,PGM=(UHL0300P,UHLNDR1P,UHL0400P),
     SERVICE=4,RMV=LX,
     CLEAN=TSTCLEAN 02980000

0033A D3E7F5F140404040 359+BTC$LX51  DC  CL4'LX51',CL4'        ' 01-00159
00342 D3E740404040000 360+  DC  CL6'LX',XL8'800000000000 000' 01-00160
00350 E4C8D3F0F3F0F0D7 361+  DC  CL8'UHL0300P',CL8'UHLNDR1P' 01-00161
00360 E4C8D3F0F4F0F0D7 362+  DC  CL8'UHL0400P',CL8'        ' 01-00162
### Registry of Motor Vehicles – UMS Programmer’s Manual

#### Chapter Five - UMS Program Control Tables

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM H V</th>
<th>02 15.16 07/15/91</th>
</tr>
</thead>
<tbody>
<tr>
<td>000370</td>
<td>00000000000000C8</td>
<td>363+</td>
<td>DC</td>
<td>XL4'0',AL4(SRV#4)</td>
<td>01-00163</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

365 * LX-WRITE OVERRIDE BREF:
* (1ST 2 FLAG BYTES=SUBFUNCTION SECURITY) 03060000

366 PCTEH LX70,PGM=(UHL0260P,UHL0400P), SERVICE=4,RMV=LX, X03070000
FLAGS=C440000000000000 03080000

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM H V</th>
<th>02 15.16 07/15/91</th>
</tr>
</thead>
<tbody>
<tr>
<td>000378</td>
<td>D3E7F7F040404040</td>
<td>367+BTC$LX70</td>
<td>DC</td>
<td>CL4'LX70',CL4' '</td>
<td>01-00163</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UHZ0002P - UMS HOST PROCESS–CONTROL TABLE

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM H V</th>
<th>02 15.16 07/15/91</th>
</tr>
</thead>
<tbody>
<tr>
<td>000380</td>
<td>D3E740404040C440</td>
<td>368+</td>
<td>DC</td>
<td>CL6'LX',XL8'C4400000000000000'</td>
<td>01-00160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00038E</td>
<td>E4C8D3F0F2F6F0D7</td>
<td>369+</td>
<td>DC</td>
<td>CL8'UHL0260P',CL8'UHL0400P'</td>
<td>01-00161</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00039E</td>
<td>4040404040404040</td>
<td>370+</td>
<td>DC</td>
<td>CL8' ',CL8' '</td>
<td>01-00162</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003AE</td>
<td>000000000000000C8</td>
<td>371+</td>
<td>DC</td>
<td>XL4'0',AL4(SRV#4)</td>
<td>01-00163</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003B6</td>
<td>FFFFFFFF</td>
<td>373 LAST</td>
<td>DC</td>
<td>XL4'FFFFFFFFFF'</td>
<td>03410000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003BA</td>
<td>00000</td>
<td>374</td>
<td>ORG</td>
<td>NUMENT</td>
<td>03420000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000</td>
<td>00000009</td>
<td>375</td>
<td>DC</td>
<td>A((LAST-FIRST)/(SYM2-SYM1))</td>
<td>03430000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000004</td>
<td>003BA</td>
<td>376</td>
<td>ORG</td>
<td></td>
<td>03440000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000</td>
<td>03450000</td>
<td>377</td>
<td>DC</td>
<td>CL8'&amp;SYSDATE',CL1' ',CL5'&amp;SYSTIME'</td>
<td>03460000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Chapter Five - UMS Program Control Tables**
RELOCATION DICTIONARY

<table>
<thead>
<tr>
<th>POS.ID</th>
<th>REL.ID</th>
<th>FLAGS</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0001</td>
<td>0C</td>
<td>000004</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td>0C</td>
<td>000336</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td>0C</td>
<td>000374</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td>0C</td>
<td>0003B2</td>
</tr>
</tbody>
</table>

LINECOUNT(65), FLAG(0), SYSPARM()
NO OVERRIDING DD NAMES
310 CARDS FROM SYSIN       0 CARDS FROM SYSLIB
452 LINES OUTPUT          23 CARDS OUTPUT

MVS/XA DFP VER 2 LINKAGE EDITOR                15:16:22  MON  JUL 15, 1991
JOB RMCJEBBH   STEP STEP010    PROCEDURE LKED
INVOCATION PARAMETERS - LIST,XREF
ACTUAL SIZE=(317440,79872)
OUTPUT IS ON VOLUME RELP02

CROSS REFERENCE TABLE

<table>
<thead>
<tr>
<th>CONTROL SECTION</th>
<th>ENTRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>ORIGIN</td>
</tr>
<tr>
<td>UHZ0002P</td>
<td>00</td>
</tr>
<tr>
<td>ENTRY ADDRESS</td>
<td>00</td>
</tr>
<tr>
<td>---------------</td>
<td>----</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LOCATION</th>
<th>NAME</th>
<th>LOCATION</th>
</tr>
</thead>
</table>

** SYMBOL IN CONTROL SECTION **

** UHZ0002P REPLACED AND HAS AMODE 24 **

** LOAD MODULE HAS RMODE 24 **

** AUTHORIZATION CODE IS 0.**

**************************** BOTTOM OF DATA ****************************
Hexadecimal Dump of PCTEH Example

AMASPZAP INSPECTS, MODIFIES, AND DUMPS CSECTS OR SPECIFIC DATA RECORDS ON DIRECT ACCESS STORAGE.
DUMPT UHZ0002P ALL 00110006

**CCHHR- 0011000401 RECORD LENGTH- 0003C8 MEMBER NAME UHZ0002P CSECT NAME UHZ0002P
000000 00000009 00000188 E4C8D9E2 E3E3D3E2 E4C8D9E2 D9C5C7E2 40404040 40404040 40404040 40404040 ..........................UHRSTTLS*
*UHRSREGS  *
  *  *
  *  *
  *  *
000020 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHUC101P*
*UHUSSDIPUHL0050P*
  *  *
000060 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHRSREGS*
  *UHL0050P  *
  *  *
000080 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHUL0200P*
  *  *
0000A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHL0065P*
  *  *
0000C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................MMRSSDIPUHL0055P*
  *  *
0000E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................MMRSSDIPUHL0055P*
  *  *
000100 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHU101CP*
  *  *
000120 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHL0055P*
  *  *
000140 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHL0065P*
  *  *
000160 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 ..........................UHL0065P*
  *  *
000180 40404040 40404040 C8E3C2F1 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 00000000 0000E4C8 ..........................HTB1  *
  *  ........UH*
<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Data 1</th>
<th>Data 2</th>
<th>Data 3</th>
<th>Data 4</th>
<th>Data 5</th>
<th>Data 6</th>
<th>Data 7</th>
<th>Data 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001A0</td>
<td>E9F0F0F0 F4D74040</td>
<td>40404040 40404040 40404040 40404040 40404040 40404000</td>
<td>*Z0004P  *</td>
<td>*</td>
<td>* .......HTB2  *</td>
<td><em>.........UH20</em></td>
<td><em>004PMRMXWAFL</em></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>0001C0</td>
<td>00000000 0000C8E3 C2F24040</td>
<td>40404040 00000000 00000000 E4C8E9F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001E0</td>
<td>F0F0F4D7 D4D9D4E7 E6C1C6D3</td>
<td>40404040 40404040 40404040 00000000 00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000200</td>
<td>00000000 D3C2F1F0</td>
<td>40404040 40404040 40404040 40404040 00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000220</td>
<td>F0D74040 40404040 40404040 40404040 40404040 4040E3E2 E4C8D3F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000240</td>
<td>0000D3C4 F0F44040 4040D3D7 40404040 00000000 00000000 E4C8D3F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000260</td>
<td>40404040 40404040 40404040 40404040 40404040 00000000 00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000280</td>
<td>D3C8F1F0 40404040 D3C94040 40400000 00000000 00000000 E4C8D3F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002A0</td>
<td>D3F0F0F6 F0D74040 40404040 40404040 40404040 40404040 0000D3C9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002C0</td>
<td>F0F44040 4040D3D7 40404040 00000000 00000000 E4C8D3F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002E0</td>
<td>40404040 40404040 40404040 40404040 40404040 00000000 00000000 D3C9F0F7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000300</td>
<td>40404040 D3C94040 40400000 00000000 0000E4C8 D3F0F0F7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000320</td>
<td>F0D7E4C8 D3F0F1F1 F0D7E3E2 E3C3D3C5 0148D3E7 F5F0F0F7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000340</td>
<td>4040D3E7 40404040 80000000 00000000 E4C8D3F0 F3F0F0D7 E4C8D3D5 C4D9F1D7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000360</td>
<td>E4C8D3F0 F4F0F0D7 40404040 40404040 00000000 00000000 D3E7F0F7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000380</td>
<td>D3E74040 4040C440 00000000 0000E4C8 D3F0F2F6 F0D7E4C8 D3F0F4F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003A0</td>
<td>40404040 40404040 40404040 40404040 00000000 0000FFFF FFFF0F7 61F1F561</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003C0</td>
<td>F9F140F1 F54BF1F6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AMA113I COMPLETED DUMP REQUIREMENTS

AMA100I AMASPZAP PROCESSING COMPLETED

*************************************************************************** BOTTOM OF DATA***************************************************************************
Internal Function Codes

UMS internal function codes are used by the Host Side Dispatcher program (UHZ0001P) to indicate which host programs, service modules, and duplicate resolution modules to invoke. The guest input program is responsible for determining which internal function code is needed and moving it to the appropriate displacement in the guest-to-host block of the common area. When control passes to the host side, UHZ0001P matches the internal function code within the guest-to-host block of the common area to the internal function code in the Program Control Table Entry Host table (PCTEH; refer to the Program Control Table section for more detail) to determine which host program to execute.

Each online UMS screen has two sets of internal function codes. One set is used with the current version of the guest and host software and the other set is available for the next version or release of software (refer to the Versioning and Naming Standards sections for more detail). In other words, each time versioning is applied to an online UMS screen, the application programmer alternates between the function codes listed in the columns below.

For example, if UPIC is versioned and the current Guest Input program references “UU14” and “UU15,” the new versioned Guest Input program would reference “UU06” and “UU07.”

The UPH screen is another example of a screen that employs two function codes at once. On the UPH screen, policy history information may be requested using either an in-state or an out-of-state license number as a key. Currently, if a request is performed using an in-state license, the Guest Input program (UGU3041P) will move a “UU08” to the guest-to-host block of the common area. For an out-of-state license request, “UU09” will be moved to the guest-to-host block of the common area. The next time UPH is versioned, “UU16” will be moved to the guest-to-host block of the common area when the new version of UPH performs an in-state license request. Similarly, “UU17” will be moved when a request using an out-of-state license is performed.

NOTE: The names and function codes below are examples only. These things change with every release of the system to minimize confusion with earlier releases of the software.
## Internal Function Codes List

<table>
<thead>
<tr>
<th>SCREEN NAME</th>
<th>FUNCTION CODE NAME 1</th>
<th>FUNCTION CODE NAME 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LH</td>
<td>LH10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LH20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LH30</td>
<td></td>
</tr>
<tr>
<td>LI</td>
<td>LI07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LI08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LI09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LI10</td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td>LI04</td>
<td></td>
</tr>
<tr>
<td>LNO</td>
<td>LD08</td>
<td></td>
</tr>
<tr>
<td>LNS</td>
<td>LD04</td>
<td></td>
</tr>
<tr>
<td>LTH</td>
<td>LI07/LB10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LI08/LB10</td>
<td></td>
</tr>
<tr>
<td>SDH</td>
<td>SDH0</td>
<td>SDH5</td>
</tr>
<tr>
<td></td>
<td>SDH1</td>
<td>SDH6</td>
</tr>
<tr>
<td></td>
<td>SDH2</td>
<td>SDH7</td>
</tr>
<tr>
<td></td>
<td>SDH3</td>
<td>SDH8</td>
</tr>
<tr>
<td></td>
<td>SDH4</td>
<td>SDH9</td>
</tr>
<tr>
<td>UPA</td>
<td>PA01</td>
<td>PA05</td>
</tr>
<tr>
<td></td>
<td>PA02</td>
<td>PA06</td>
</tr>
<tr>
<td></td>
<td>PA03</td>
<td>PA07</td>
</tr>
<tr>
<td></td>
<td>PA04</td>
<td>PA08</td>
</tr>
<tr>
<td>UPCR</td>
<td>UU04</td>
<td>UU12</td>
</tr>
<tr>
<td></td>
<td>UU05</td>
<td>UU13</td>
</tr>
<tr>
<td>UPH</td>
<td>UU08</td>
<td>UU16</td>
</tr>
<tr>
<td></td>
<td>UU09</td>
<td>UU17</td>
</tr>
<tr>
<td>UPIC</td>
<td>UU06</td>
<td>UU14</td>
</tr>
<tr>
<td></td>
<td>UU07</td>
<td>UU15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>UPOI</td>
<td>UU10</td>
<td>UU18</td>
</tr>
<tr>
<td>UPTH</td>
<td>UU19</td>
<td>UU20</td>
</tr>
<tr>
<td>URI</td>
<td>URI1</td>
<td>URI5</td>
</tr>
<tr>
<td>URI2</td>
<td>URI6</td>
<td></td>
</tr>
<tr>
<td>URI3</td>
<td>URI7</td>
<td></td>
</tr>
<tr>
<td>URI4</td>
<td>URI8</td>
<td></td>
</tr>
<tr>
<td>URSN</td>
<td>UR04</td>
<td>UR07</td>
</tr>
<tr>
<td>URSR</td>
<td>UR02</td>
<td>UR05</td>
</tr>
<tr>
<td>URSV</td>
<td>UR03</td>
<td>UR06</td>
</tr>
<tr>
<td>URVN</td>
<td>UR08</td>
<td>UR16</td>
</tr>
<tr>
<td>USH</td>
<td>SH03</td>
<td>SH06</td>
</tr>
<tr>
<td>SH04</td>
<td>SH07</td>
<td></td>
</tr>
<tr>
<td>SH05</td>
<td>SH08</td>
<td></td>
</tr>
<tr>
<td>UVH</td>
<td>VH01</td>
<td>VH03</td>
</tr>
<tr>
<td>VH02</td>
<td>VH04</td>
<td></td>
</tr>
</tbody>
</table>
Feature Summary -- LXTABLE Processing

The UMS subsystem has incorporated an automatic screen data handler referred to as “LXTABLE Processing,” which is located in its GUEST software. This processing mechanism is invoked transparently to the application program(s) by the Control/Dispatch mechanism based on the XLATE=<lx-tbl-name> parameter in the System Control Table. The value <lx-tbl-name> is the linkedit name of the assembled LXTABLE. On input and output operations, it converts data from screen to internal format and vice versa. On input only, it will perform the value edits and/or specific content/cross field edits requested. If edit errors occur on input, control never passes to the application but the screen is rewritten, with error messages, for correction. The input-edit error-message cycle continues until all fields pass the edits. At that point, control is passed to the designated application program with the screen-data in the Common Area.

LXTABLE processing has two components, the table-driven ‘LX-Program’ (UGZ0006P) and a macro generated, application specific ‘LX-table’ linked as a load module. This mechanism’s use is required in all UMS applications to standardize field editing and screen formatting functions. As the program UGZ0006P serves all applications, it does reduce programmer effort in these areas speeding application development and reducing maintenance time. Its basic design is rooted in a product designed and implemented for the ALAR RA function, but the UMS version is quite different.

The LX-table relates the data fields of the user’s commarea with the defined screen-dsect for the application. This table is created by the programmer and defines all the involved data fields. The entries in the table for each field pair may specify certain standard edits, including cross-field checks and values. In this discussion, the term LXTABLE refers to the composite effect of the two components.

The LXTABLE mechanism depends in part, on the presumption that each element has two control fields associated with it. These fields are each 1 byte long and are named ZFLD and TFLD. The ZFLD byte is used to contain commands relative to the field passed between the application and the LXTABLE mechanism. The TFLD is seldom used and is a provision for passing an optional override attribute. The derivation of the names ZFLD and TFLD is rooted in the ALAR RA implementation, and the mnemonic implications of these names (if any) is not known.

General data elements may be located anywhere in the UMS common area, but the structure requires that each data element be immediately preceded by the ZFLD and...
TFLD bytes. This is very dissimilar from the ALAR RA implementation which uses 3 lists of fields. The processing supports the use of output-only (as opposed to general) data elements. These elements are always located in the host interface block and are not preceded by the ZFLD/TFLD set.

LXTABLE provides automatic field conversion services for the application. This relieves the application of this coding and insures that they are done in a consistent manner. Complete instructions for use are given in the Detail Description portion of this chapter. The following are some of the types of field conversion supported. See the Edit Type Table on page 107 for a full listing:

1. **Date:** on input, dates are analyzed for proper syntactical format (mm/dd/yy, mm/dd/yyyy, mmddyy, mmddyyyy, yyyy/mm/dd) and converted to standard halfword binary (WAASDATE, # of days away from 1/1/41) format. The various components of the date are also checked for legitimacy. On output, the standard halfword binary dates are converted to mm/dd/yyyy format. The null-date (x'8000') is properly converted.

2. **Month/Year dates:** these dates are converted to and from the RMV standard internal format for month/year dates, with proper field editing. The most common use for this type of field is registration expiration dates.

3. **Zipcodes:** zipcodes are converted to and from standard RMV internal form.

4. **Case:** for input fields, LXTABLE automatically performs lower to upper case translations. This is particularly important in the UMS guest environment where it is not possible to require that the terminals be configured for upper-case only. Further, the same process strips nonstandard display characters which may be included by some non-IBM 3270 type devices.

5. **Erase-EOF:** automatically converted to the null value for the defined field type.

6. **Numeric:** on input, fields with a generic type of numeric are edited for numeric content (or omitted), and treated as though they had been entered with right justification and left zero fill. On output, they are formatted according to several possible standardized structures.

7. **Numeric Conversion:** fields on the screen with a generic type of numeric clearly must contain EBCDIC data. LXTABLE allows the programmer to have his internal data definitions in one of several supported formats. The programmer may define the internal numeric fields as packed, binary or zoned and LXTABLE will automatically convert the screen fields to/from these formats.

LXTABLE provides a generalized VALUE editing scheme for screen fields. This means that the programmer, in the LXTABLE itself, may specify lists of values which are allowed (or disallowed) for each field. LXTABLE will force the conformation to these
specifications. This facility provides a generic error message for violations, or allows the option of specifying a specific text for each field. The EXIT facility allows the user to write specialized edits that may be used on one or more screen fields.

On output, LXTABLE accepts commands (in the ZFLD byte) relative to each field. The commands and their values ( ) are:

1. ERROR: (E) highlight the field, position the cursor to the field if it is the first such field on the screen.
2. TRANSMIT: (X) the field is overlaid on the screen with its current content in the commarea, after proper conversion.
3. ERROR/TRANSMIT: (H) a combination of 1 and 2.
4. RESET: (R) the field attribute is set to its default value from the map definition.
5. TRANSMIT/CURSOR: (K) the field is overlaid on the screen with its current content in the commarea, after proper conversion. The cursor is positioned to the field if it is the first such field on the screen.

After processing these commands, LXTABLE clears the ZFLD byte.

The ZFLD byte for unprotected input fields is used to indicate to the application program if the field in question has been logically changed (value = ‘C’). Logically changed does not mean the MDT is set, but rather means that, after all involved conversions, it has a different value than the value currently stored in the commarea. In the ZFLD notification of change, LXTABLE uses a stepdown technique of indicating change in current iteration (‘C’) and change in previous iteration (‘P’). All of these notifications are superseded by any output command for the field which is issued by the application.

The TFLD serves two functions. On input, if the option PASSMDT=YES is set, the presence of the MDT for the field is shown by B'00000100' in the TFLD. Absence of the MDT is shown as X'00'. On output, any value in the TFLD is sent to the screen as the attribute byte for this iteration.

LXTABLE contains an option (AUTORES) which allows the reset-to-default attribute option to be the default for all transmit operations.

LXTABLE incorporates a concept known as field classes. The classes are DATA, KEY, and KEY-SUPPORTIVE. The purpose of this concept is to allow a screen to be partially edited. In essence, if fields are put in the class of KEY -- AND ARE LOGICALLY CHANGED -- the editing of DATA fields which follow on the screen is bypassed. This technique is important for situations where the operator may get sufficiently wound-up in errors to decide that his only option is to alter a key field and essentially restart a transaction.
LXTABLE processing is declared for a function in its UMS System Process Control Table Entry, Guest (PCTEG). The exact parameter is:

   XLATE=<lx-tbl-name>

Providing the 8-character LX-table name (<lx-tbl-name>), activates LXTABLE processing at the appropriate times. As part of the NDR implementation, a technique of invoking the UMS GUEST LXTABLE processor from an ALAR application was developed. This technique is not transparent to the application (as in the UMS GUEST version), but requires the addition of a few statements to the application program.

The LXTABLE exception EXIT processing facility is a mechanism which allows the programmer to augment the LXTABLE processing on a field by field basis in designated circumstances such as edit fail, edit pass, or every time. This EXIT facility allows the execution of user code within the LXTABLE input process. This code may be used to issue errors not normally recognized by LXTABLE or to bypass errors normally recognized. These conditions may be recognized with complete access to the commarea, as well as access to the map data (normally not available with LXTABLE processing). Note that this option is only to be used in exceptional circumstances. If the editing to be done will be required in other places in the system, then a standard edit will be produced to use with the EDIT parameter. This exit facility is further described in the second part of this chapter.

The field types defined below are supported.

<table>
<thead>
<tr>
<th>type</th>
<th>minimum</th>
<th>maximum</th>
<th>GSA data-type</th>
<th>MAP data-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>1</td>
<td>79</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>num</td>
<td>1</td>
<td>15</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>packed</td>
<td>1</td>
<td>15</td>
<td>P</td>
<td>Z</td>
</tr>
<tr>
<td>binary</td>
<td>1</td>
<td>9</td>
<td>FL4 or F</td>
<td>Z</td>
</tr>
<tr>
<td>date</td>
<td>10</td>
<td>10</td>
<td>HL2 or H</td>
<td>C</td>
</tr>
</tbody>
</table>

The LXTABLE software processes fields in the order in which they are assembled in the LXTBL entries in the LX-table module. These entries should be defined left to right, top to bottom in relation to the screen. If they are not, processing will occur in a proper manner but an error message posted will refer to the first processed field, not necessarily the first on the screen with an error.

Fields determined to be changed have their flag-byte value first set to an intermediate value (which is C'*') until all error-resolution is complete, and are then set to the proper value of C'C' before the application program is invoked. In addition to being intensified, the error fields have their MDT forced on when the error screen is shown, to insure that they are reentered.
The LXTABLE software maintains two levels of field change detection. The basic level is finding the MDT that is turned on. Once this is detected, the software compares the value in the field (after it is syntax checked) with the value currently in the GSA area. If they are equal, the field is treated as though it had not been changed. For these purposes, a numeric field which has been erase-EOF'd is treated as if a value of 0 (zero) had been entered. Note that when comparing numeric fields (zoned or packed), if the GSA data area does not contain a valid sign code, the comparison is treated as not-equal (field changed).

There is a potential case where a field may be flagged as changed but logically have the same value as prior to the change. This occurs if (1) a legitimate change is detected on the field, (2) in the same demap operation an error is detected in another field, and (3) during the correction of the error the user changes the first field back to its original value.

If the application program indicates a field with error, and does not provide an attribute byte, the mapping software will supply a default attribute. The value will be intensified, unprotected, MDT-on. If the field is numeric (in the LXTBL definition) this attribute will be set as well.

The fields corresponding to mapname, maplength and cursor-position in the guest common-area should not be changed. If the map/demap software detects a changed value in maplength, it will assume that the user has prepared the correct map for transmission and will perform no output gsa-map interactions. This is an override feature available to the application which must be exercised with EXTREME care as it is possible to create an unresolvable error-conflict situation after the next receive operation.

**UMSLXTBL Quick Look**

To implement LXTABLE processing for an application, one must add the name of the LX-Table to the Guest PCTEG in the form of the parameter

```
XLATE=<lx-tbl-name>
```

To create the LX-Table itself the CICS map and its dsect must be created along with the GSA Common storage definitions the application program will use. With these available the LX-Table may be assembled. It requires one TYPE=START macro and, for each screen-field / GSA Common field pair, one TYPE=DETAIL macro. Descriptions of the parameter values is in the following section, Detail Description. When items are enclosed in `{ }`, choose one of them.
<name>  UMSLXTBL TYPE=START,MAPNAME=<mapname>,AUTORES=YES/NO, PASSMDT=YES/NO, LEVEL0=YES/NO/<null>

UMSLXTBL TYPE=DETAIL,MAPFLD=<mfldbname>,
{ GSAFLD=<gfldbname>, | HOSTFLD=<hfldbname>, }
OPTION1=<null> | LZFILL | RTJUST,
KEYFLD=<null> | 0 | 1 | 3 | 4,
VALUE=(<vtype>,<valuelst> | R<vtype>,<value-pairs>),
VALERROR=<error-num>,
EDIT=(<editype><edit fld lst>),
REQUIRE=YES | NO,
EXIT=(<exitparms>)

**Detail Description**

*The LXTABLE Macro*

This section discusses the parameters of the LX-Table macros in detail with some examples. The LXTABLE relates mapfields to storage fields and specifies certain kinds of automatic edits and value checking that may be required. The data generated in the table allows the MAP/DEMAP Module to move the data from the mapped-in screen to common storage or vice versa with the editing requested and certain data-type conversions (eg. edited date <---> internal half-word format date). The default value for a parameter will be in **BOLD** type. APPENDIX L has a sample generation of a small LX table showing the various parameters and features in use. Part I of the appendix is the assembly listing to illustrate the macro expansions and code generation. Part II is a hex dump of the load module from the assembly (and link). These examples are marked to show where the generated code appears in the load module. Fragments of code are included as examples with this discussion.

The LX-Table load module will always have the csect <lx-tbl-name> as the first csect. If the VALUE or EDIT parameters are specified a second csect, always called CSECT2, will be generated in the load module.

In the various illustrations, <null> means the parameter was omitted (or specified as, for example, LEVEL0=, or LEVEL0=""). The vertical bar | separates the valid parameter values. The <null> represents the omitted parameter, or value, as mentioned previously. A bold value is the **default** parameter value.
**The UMSLXTBL TYPE=START**

The first invocation, or macro in the table, must be:

```
<name>  UMSLXTBL TYPE=START,MAPNAME=<mapname>,AUTORES=YES/NO,
        PASSMDT=YES/NO,
        LEVEL0=YES/NO/<null>
```

*<name>* is the name of the LX-Table to be created, 8-characters long

**TYPE=START** generates the LX-Table header.

*<mapname>* is the name of the related map, 7-characters long

The above are required on TYPE=START and are **not** allowed for TYPE=DETAIL.

**LEVEL0=**YES | NO allows or suppresses an mnote showing the type of edit that was requested for the individual fields.

**PASSMDT=**YES | NO pass or not pass the modified data tag through to the application program

**AUTORES=**YES | NO reset or not reset screen attributes to the map default
Sample expansion, TYPE=START

The field generations for TYPE=START are as follows:

```
1227 *

1229 UGL0260T UMSLXTBL TYPE=START,
      MAPNAME=UGL0260,
      LEVEL0=NO
1230+UGL0260T START 0
1231+    PUSH PRINT
1232+    PRINT OFF
1249+    POP PRINT
1250+    USING MAPINP,R4
1251+    USING
000000 E4C7D3F0F2F6F0E3 1252+    DC CL8'UGL0260T'
000008 E4C7D3F0F2F6F0F0 1253+    DC CL7'UGL0260'
00000F F0F661F1F861F9F1 1254+    DC CL8'06/18/91'
000017 40 1255+    DC CL1' '
000018 F2F34BF5F2      1256+    DC CL5'23.52'
00001D 00000000 1257+    DC XL4'0'
000021 00 1258+    DC BL1'00000000' FLAGS
000022 000000 1259+    DC XL3'0' RESERVED
000025 FFFF 1260+    DC XL2'FFFF' TEMP TRLR
```

Note: this code generation is streamlined for ease of viewing with proper code output.

Line 1252 is the name of this LX-Table. Line 1253 is the related CICS map(set) name. Note that the date and time of assembly are assembled into the header for documentation purposes (lines 1254 and 1256). The blank in line 1255 is for date-time readability. In line 1258, the field of FLAGS, the bits are used as follows (left to right):

1....... reset MDTs (AUTORES=YES)
.1....... pass MDTs through (PASSMDT=YES)
..000000 unused

The fields in lines 1257 and 1259 are unused. In line 1260 the XL2'FFFF' in this and the expansions of UMSLXTBL TYPE=DETAIL, exist only if ‘this’ invocation of the macro is the last in the whole table. If it is not the ‘last,’ the expansion of the succeeding macro overlays it. If it is the last, the MAP/DEMAP Module knows it just saw the last set of fields to process.
The **UMSLXTBL TYPE=DETAIL**

The fields and their edits are described below. The field moves and edits proceed in the order in which the field pairs are assembled into the LX-Table. You therefore control the processing order of the screen fields. When items are enclosed in `{ }`, choose one of them.

```plaintext
UMSLXTBL TYPE=DETAIL, MAPFLD=<mfldname>,
    { GSAFLD=<gfldname>,| HOSTFLD=<hfldname>, }
OPTION1= <null>| LZFILL| RTJUST,
KEYFLD= <null>| 0| 1| 3| 4,
```

The 5 following parms are optional and related to edits for the field. They are discussed in groups after the parameters above.

```plaintext
VALUE= <null> | (<vtype>,<value> | 
    R<vtype>,<valuepair>)
VALEROR= <null> | <error-num>,
EDIT= <null> | (<editype>,<edit fld list>),
REQUIRE= NO | YES,
EXIT= <null> | (<excsect>,<exreas>,<exinvn>)
```

**MAPFLD, GASFLD, HOSTFLD**

The TYPE=DETAIL describes a CICS map field-name GSA Common data field-name pair between which data will be moved with needed data-conversion and optional edits. The CICS map field-name is `<mfldname>` and the MAPFLD parameter is required. The GSA Common field may be in either ‘program storage’ or in HOST related storage. If it is in ‘program storage’ use the GSAFLD parameter with `<gfldname>` as the field name. For HOST related storage, use the HOSTFLD parm with `<hfldname>`. If HOSTFLD = `<hfldname>` is specified, the field-pair is restricted for storage-to-screen use only (&OUTONL = ‘1’). The ‘program’ field name is placed into the internal macro variable, &INTFLD. Data-field types are checked.

Allowed for MAPFLD are T'= C | Z.

Allowed for ‘program’ fields are T'= C | Z | F | H | P | X | G.

If the type is ‘G’, and L'= 2, the type is set to ‘H’, or if L''= 4, then type is set to ‘F’. Anything else is an error. The valid type-pairs for `<mapfld><intfld>` are ‘ZZ’, ‘ZP’, ‘CC’, ‘ZF’, ‘CH’, and ‘CX’. The general edit rules and internal macro field settings are as follows:
'ZZ'  Both zoned numeric - max 15 digits long, field lengths to be equal

&NUMFLD  '1'  &RTN  '000'
&VALLOW  '1'  &VALHI  L'<mapfld>
&IFVALOK  '#'  &VALHDR  'PL&WRKLEN'
&WRKLEN  (L'<mapfld>/2)+1

'ZP'  Zoned numeric screen, packed in 'program' L'&INTFLD to be LE 8, L'&INTFLD+L'INTFLD-1 to be GE L'&MAPFLD

&NUMFLD  '1'  &RTN  '004'
&VALLOW  '1'  &VALHI  L'<mapfld>
&IFVALOK  '#'  &VALHDR  'PL&WRKLEN'
&WRKLEN  (L'<mapfld>/2)+1

'CC'  Both character fields with fields to be of equal length and LE 79 characters long.

&NUMFLD  '1'  &RTN  '008'
&VALLOW  '1'  &VALHI  L'<mapfld>
&IFVALOK  'C'  &VALHDR  'CL&WRKLEN'
&WRKLEN  L'<mapfld>

'ZF'  Zoned numeric screen field, length LE 9, and L'&INTFLD to be EQ 4

&NUMFLD  '1'  &RTN  '012'
&VALLOW  '1'  &VALHI  L'<mapfld>
&IFVALOK  '#'  &VALHDR  'FL4'
&WRKLEN  '4'

'CH'  Screen date to internal fmt (no value editing allowed), L'&MAPFLD to be 7 or 10, L'INTFLD to be 2

&NUMFLD  <null>  &RTN  '016' (L'10),
           &RTN  '020' (L'7)
&VALLOW  '4'  &VALHI  '4'
&IFVALOK  'X'  &VALHDR  'AL2(X'"
&WRKLEN  '2'  &DATEFLG  '1'

'CX'  Screen zip-code to internal fmt, L'&MAPFLD to be 10, L'INTFLD to be 6
OPTION 1

RTJUST and LZFILL (for numeric only) are edited. These actions are applied upon output only. They have no effect at screen-input processing time. Type flags are checked and, if &WARN NE 'N', macro emits MNOTE,0 for edit specified, if any.

If &VALUE NE <null>   &VALFLAG '1'
If &EDIT  NE <null> &EDTFLAG '1'

&CHKMAP = 'M$&MAPFLD'

&XYZ is set to a concatenation of all flags

&OP1FLAG&ZIPFLAG&EDTFLAG&VALFLAG&OUTONL&EXPIFLG&DATEFLG &NUMFLG

Code generation now occurs for the basic part of the macro and, if there are no value or edit clauses, the macro expansion is completed. A general layout follows with some actual examples in the following section.

Code generated is:  in 'main' csect

1   &CHKGSA   DC AL1(<dns>),AL1(&RTN)
2   &CHKMAP   DC BL1'&XYZ'
3       DC BL1'&KEYF&OP1FLAH'
4       DC AL2(&INTFLD-&BASE-2)
6       DC AL2(&MAPFLD-MAPINP-3)
5       DC AL1(L'&INTFLD-1),AL1(l'&MAPFLD-1)
6       DC AL4(<v-array>) -- value array, if used
7       DC AL4(<e-array>) -- edit array, if used
8       DC XL2'FFFF'

1. First is one byte of displacement (<dns>) to the next table section, either another field or EDIT/VALUE fields for this one. Points to the X"FFFF"
only if this the end of the table. The X'FFFF' is overlaid by the continuing or the next macro expansion.
Second is one byte of field-pair type.

2. One byte of flags as follows:

1........ Left Zero Fill asked
.1....... Zip Code fields
..1..... Edit control block present
...1.... Value control block present
....1... Out only field-pair
.....1.. Expiration format date
.......1. Internal-External date fields
..........1 A numeric value pair

3. One byte of flags as follows:

0000.... Unused
....111. KEYFLD value in binary
.......1 Right Justify asked

4. Two bytes for program-field displacement from either UGCOMMON or UHCITEXT. Then two bytes of map-field displacement from MAPINP.

5. One byte = length of 'program'-field -1.
One byte = length of map-field -1.

6. A pointer to the value array (if specified in a VALUE parameter) in CSECT2 of the load module.

7. A pointer to the edit array (if specified in an EDIT parameter) in CSECT2 of the load module.

8. Two bytes, X'FFFF', as an end of table flag. If there is another macro after this one, its expansion overlays the X'FFFF'.

KEYFLD

The form of the parameter is:
The ability to designate key-fields applies a further control to the automatic edits. If a field designated as “key” is found to be changed (and passes any edits), all other edits for non-key fields will be skipped. The assumption is that where key data is changed, the operator wishes to switch to a different “base entity” such as a different person or vehicle. The value of “4” will direct that the edit be carried out even in the face of a key-field change. The value of “3” gives the application the ability to dynamically designate the field as key or non-key at run-time. If the first byte of the USER portion of the Commarea is X'04', the “3-fields” will be edited as key. If the first byte is some other value, they are treated as ordinary fields.
Value Checking

VALUE, VALEROR

The value checking mechanism enables one to allow or disallow a single value, a series of individual values, or a range of values through the VALUE= parameter in the UMSLXTBL macro. The 9-digit VALEROR= parameter value directs the issuance of a specific error message if the value edit fails. If the value edit fails and no VALEROR parameter is given, a generic error message issues. All the code generated for value editing is assembled into the csect called CSECT2. The LX-Table load module will always have the csect <name> and optionally, CSECT2 if value checking and/or editing is specified. MNOTEs are issued for any edit failure in the macro. The parameter is coded as follows:

The presence of a 9 numeric digit <errornum> in the VALEROR= clause, causes &ECODE to be set to '1' (else it is '0').

If &IFVALOK is 'X', &SIGN is set to '1', else '0'.

VALUE=<null> | (<vtype>,<valuelst> | R<vtype>,<value-pairs>)

Where:  <vtype> = ALLOW | DISALLOW
<valuelst>    = <value>,<value> ... as required
<value-pairs> = <value2>,<value2> ... as required
<value2>      = <value>,<value> a pair for a range
    <value> = any value the field may assume

The VALUE clause may not be used on output-only fields (&HOSTFLD that causes &OUTONL to be '1') or on fields where &IFVALOK is left as 'N'. The characters in the value string are edited for correctness as numerics or a hex string as the case may be. An MNOTE issues in case of error. Two internal macro values are set based on <vtype>. The VALUE clause may be specified with the EDIT clause in the same TYPE=DETAIL macro. Note that in such a case, the VALUE clause edits are applied before those of the EDIT clause.

WARNING: An input value of 0 (zero) or blanks will be passed through to the application regardless of the value clause's sub-parameters.
&COMPTYP &VALTYPE Value

<vtype> = ALLOW '0' '0' single
DISALLOW '1' '0' single
R<vtype> = ALLOW '0' '1' range
DISALLOW '1' '1' range

**Code generated is:** In 'main' csect

DC AL4(<v-array>) --> to value list in csect2

**Code generated is:** In CSECT2

DC FL4'&VALEROR' error msg number (if any)
<vlabel> DC BL1'&ECODE&SIGN&COMPTYP&VALTYP' flags
DC AL1(<actual field length>)
DC AL2(<number of list elements>)
DC <elements>
&SYSECT CSECT return to 'main' csect

<elements> = sets of the data type, the true field length and a value that represents the value(s) in the list.

For example CL3'C&C' CL5'RMV' ZL4'789'

If &VALEROR is <null>, &ECODE is '0' and the FL4 field is not generated. If &VALEROR is given and a value edit fails, that error message will be displayed on the screen with out application program intervention. If VALEROR is omitted, a generic message is sent.
Edit Checking

EDIT, REQUIRE

The EDIT clause routines perform services such as check-digit validation, ensure numerics, town-code conversions, date checks and conversions, and non-database cross-field checking. One may designate the field as required, REQUIRE=YES, to produce an error message if the field is not supplied. Code generation builds control blocks in CSECT2 of the LXTABLE. The EDIT and VALUE clauses may be specified in the same TYPE=DETAIL macro. In such a case, the VALUE clause edits are applied those of the EDIT clause.

The form of the parameters are:

REQUIRE=NO | YES if Yes and no data, error
if YES, &REQ = 1 ( else = 0 )

EDIT=(<editype><edit fld lst>)

<editype> a literal from the Edit Type Table on page 5-18 that tells UGZ0006P(map/demap pgm) what to do

<edit fld lst> = <fldn> | <fldn>,<fldn+1> ...

<fldn> etc is the name(s) of the field(s) to be edited (the table shows the requirements)

Code generated: in csect CSECT2

<E-label> DC AL1(<e-label - E-label) length of this array
DC AL1(&REQ&MODE) &MODE unused, = 0
DC AL2(&TYPE) edit to do, 0-19
DC AL4(<lxptr>) --> block in main table
<E-label> DC 0AL1 label for length calc
**Edit Type Table**

Note: The number of fields required is absolute, too few or too many errors out the macro with an MNOTE. If the # of fields required is 1, the edit applies to the current macro being expanded. If more than 1 field is required, they must be in the current macro or already expanded. If any of the fields come later, the macro errors out. The fields must have the type and length specified or the macro fails.

<table>
<thead>
<tr>
<th>Edit Type</th>
<th>Literal</th>
<th>Description</th>
<th># Fields Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LICNO</td>
<td>MA lic #, 9-byte</td>
<td>1 CL9 license number</td>
</tr>
<tr>
<td>1</td>
<td>VIN</td>
<td>VIN, 17 byte, chkdgt</td>
<td>1 CL17 the VIN</td>
</tr>
<tr>
<td>2</td>
<td>TCODE</td>
<td>Town Code</td>
<td>1 PL2 internal town code</td>
</tr>
<tr>
<td>3</td>
<td>ICODE</td>
<td>Insurance Code</td>
<td>1 PL2 internal Co. id.</td>
</tr>
<tr>
<td>4</td>
<td>REGNO</td>
<td>Registration</td>
<td>3 CL3 plate type, CL1 plate color, CL7 registration #</td>
</tr>
<tr>
<td>5</td>
<td>MADDR</td>
<td>Optional Address</td>
<td>5 (see type 6)</td>
</tr>
<tr>
<td>6</td>
<td>RADDR</td>
<td>Required Address</td>
<td>5 CL20 address line 1, CL20 address line 2, CL15 city/town, CL2 state, XL6 zip code</td>
</tr>
<tr>
<td>7</td>
<td>MLICN</td>
<td>License Number, 9-byte</td>
<td>2 CL9 license number, CL2 state</td>
</tr>
<tr>
<td>8</td>
<td>TITLE#</td>
<td>Title Number</td>
<td>1 CL8 the title number</td>
</tr>
<tr>
<td>9</td>
<td>PDATE</td>
<td>Policy Date Ranges</td>
<td>2 H effective date, H expiration date</td>
</tr>
<tr>
<td>10</td>
<td>PNAME</td>
<td>Person Name</td>
<td>3 CL16 last name, CL12 first name, CL8 middle name</td>
</tr>
<tr>
<td>11</td>
<td>ILOCN</td>
<td>Insurance Co. Location</td>
<td>2 PL2 Co. code, internal, PL2 town code, internal</td>
</tr>
<tr>
<td>12</td>
<td>MLINC2</td>
<td>License #, 25-byte</td>
<td>2 CL25 license #(non MA), CL2 state</td>
</tr>
<tr>
<td>13</td>
<td>SOCNO</td>
<td>Social Security #</td>
<td>1 CL9 SSAN (CL / ZL)</td>
</tr>
<tr>
<td>14</td>
<td>MONEY</td>
<td>Money edit</td>
<td>1 CL3 to CL21, money field</td>
</tr>
<tr>
<td>15</td>
<td>LICNO/S</td>
<td>Lic # w/'S' allowed</td>
<td>1 CL9 license #</td>
</tr>
<tr>
<td>16</td>
<td>MLICN2</td>
<td>License, 25-byte w/</td>
<td>2 CL25 license #</td>
</tr>
<tr>
<td></td>
<td>/NOMA</td>
<td>no ST default</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>AGE1511</td>
<td>Age Edit, dob/permit</td>
<td>1 H dob, internal fmt</td>
</tr>
<tr>
<td>18</td>
<td>AGE1600</td>
<td>Age Edit, dob/license</td>
<td>1 H dob, internal fmt</td>
</tr>
<tr>
<td>19</td>
<td>MLICN2/XX</td>
<td>Lic # 25 byte</td>
<td>2 CL25 license #</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w/ XXnolicense</td>
<td></td>
</tr>
</tbody>
</table>

Chapter Six - LXTABLE Processing
User Exit - LXTABLE

An LXTABLE exit is user-code included as part of the assembly and link of an
LXTABLE. This feature provides processing of editing criteria, which are NOT an
integral LXTABLE function, during the de-map phase of LXTABLE processing (before
invocation of the application program). This supports the simplification objective of
keeping screen edits outside the application program.

These exits are not designed to (and are not to be used to) replace (or augment) standard
procedures which occur in many places. They are intended to be used in “unique”
circumstances. That means that if some type of common field not yet accounted for by
LXTABLE evolves, it is not to be processed with an exit BUT, through an addition to the
standard LXTABLE processing.

Examples of the exit code are provided as a code-fragment on page 111 and as a part of
an assembled table that begins on page 117.

EXIT Specification:

The specification of these exits is a part of the UMSLXTBL macro. The involved
argument is:

EXIT=(<exname>,<exr>,<exc>).

<exname>  is the 1-8 character csect name assigned to
the exit routine

<exr> = <null> | <exreason>
<exreason>  is a 2 hex-digit (or 8 binary-digit)
expression of why/when the exit is to be
invoked, to wit:

X'10' always (unless the data fails
conversion, eg. non-numeric in numeric
field), before value and editing
processing.
X'08' after failure of an edit operation
X'04' after passing an edit operation
X'02' after failing a value-check operation
X'01' after passing a value-check operation

<exc> = <null> | <excode>
<excode>  is an unsigned halfword in the decimal
number range of 1 - 65535, which uniquely identifies the exit invocation.

The value of <exreason> is not checked at generation time to determine if the user has specified values other than the above. If no value is specified, an automatic value of ALWAYS (X'FF') is generated. Specification of a value of zero is detected as an error. If a 2 digit value is specified, the digits must be HEX (0-9, A-F), if a 8 digit value is specified, the digits must be BINARY (0-1). No other possibilities are allowed.

The <excode> value (ranging between 1 and 65535 decimal) is intended to allow the user's exit to be invoked relative to more than 1 map-field and to be able to differentiate which invocation is current. Each potential value may occur only once in any single LXTABLE. The field may be omitted, in which case a default value is assigned. The first default value is 65535, the second 65534, etc.

If it is desired to enter an exit for multiple reasons, the desired reasons may be OR'ed together to form a composite reason. When the routine is entered, it is informed of the specific reason for this entry (as opposed to any composite), as well as a history of its entry for this field within this LXTABLE invocation (EXITWHYS).

Examples:

* EXAMPLE: NUMERIC TO PACKED, OUTPUT MUST BE LONG-ENOUGH

    UMSLXTBL MAPFLD=MNUM2,GSAFLD=GNUMP,OPTION1=LZFILL,
    EXIT=USER01

  *

* EXAMPLE: DATE, INPUT MUST BE CL10, OUTPUT MUST BE H OR HL2

    UMSLXTBL MAPFLD=MDATE,GSAFLD=GDATE,EDIT=AGE1600,
    VALUE=(ALLOW,8000,0000,FFFF,7FFF),EXIT=(USER02,01,99)
Exit Environment:

The exit is presented with the GUEST common area and an exit argument block. It can make any evaluation it desires and indicate three possible actions:

a. force a specific error condition
b. force no-error condition
c. continue with normal processing

In all cases, normal processing would be (1) no-error, or (2) some error detected by normal LXTABLE processing.

On entry:

(R12) holds the address of the exit
(R9) holds the address of the GUEST common
(R8) holds the address of the exit argument block.

The exit argument block is as follows:

EXITDATA DSECT
EXITCODE DS AL2 UNIQUE IDENTIFIER CODE
EXITWHYS DS XL1 SUMMATION OF ENTRY REASONS
EXITRETN DS AL4 RETURN ADDRESS
EXITWHYX DS XL1 REASON FOR THIS ENTRY
EXRVALOK EQU B'00000001' VALUE IS OK
EXRVALNG EQU B'00000010' VALUE IS NOT ACCEPTABLE
EXREDTOK EQU B'00000100' EDIT IS ACCEPTABLE
EXREDTNG EQU B'00001000' EDIT IS NOT ACCEPTABLE
EXRALWAY EQU B'00010000' ALWAYS ENTER UNLESS FMT ERR
EXITUSER DS 0XL5 USER PARMS
EXITACTN DS XL1 USER ACTION REQUEST
EXANOACT EQU B'00000000' CONTINUE, NO ACTION
EXANOERR EQU B'00000001' CONTINUE, NO ERROR
EXAWIERR EQU B'00000100' CONTINUE, POST ERROR
EXITEROR DS XL4 USER ERROR CODE

EXITCODE the value of <excode>
EXITWHYS a running OR of the reasons this exit has been invoked for this field in this LXTABLE invocation
EXITRETN return address
EXITWHYX the reason for this invocation <exreason>
EXITACTN  user specified action
0 continue with normal processing-
default
1 continue with no-error
2 continue with error
EXITEROR  error-code to post if EXITACTN=2.
defaults to a generic error-code.

User Exit Example - Code Fragment

Example #1

* USER EXITS:
  DROP
UGLTEST1 CSECT
  USING UGCOMMON,R9           COMMON AREA
  USING EXITDATA,R8           EXIT DATA
  USING USER01,R12            ROUTINE BASE

* THIS EXAMPLE WILL FORCE NO-ERROR IF THERE IS AN EXISTING ERROR
* AND WE HAVE BEEN ENTERED FOR THE REASON OF AN ERROR. IN ANY
* OTHER CASE, NORMAL FLOW WILL BE OBSERVED

  USER01    DS    0H
  OC    UGGERRCD,UGGERRCD     PREVIOUS ERROR?
  BZ    U19999               NO
  TM    EXITWHYX,EXRVALNG+EXREDTNG   REASON=ERROR?
  BZ    U19999               NO
  MVI   EXITACTN,EXANOERR       FORCE ERROR
  U19999 ICM   R14,B'1111',EXITRETN    GET RETURN ADDRESS
  BR    R14                     EXIT

Example #2

  DROP
UGLTEST2  CSECT
  USING UGCOMMON,R9          COMMON AREA
  USING EXITDATA,R8          EXIT DATA
  USING USER02,R12           ROUTINE BASE

* THIS EXAMPLE WILL FORCE AN ERROR IF GDATE=X'7FFF' AND GCHAR=
* ALL '$'. IN ANY OTHER CASE, NORMAL FLOW WILL BE OBSERVED
USER02    DS    0H
CLC   GDATE,=X'7FFF'      7FFF?
BNE   U29999              NO
CLI   GCHAR,C'='$'          1ST BYTE OF GCHAR=$?
BNE   U29999              NO
CLC   GCHAR+1(L'GCHAR-1),GCHAR   REST=$?
BNE   U29999              NO
MVC   EXITEROR,=F'123456789'   SET ERROR-CODE
MVI   EXITACTN,EXAWIERR        FORCE ERROR
U29999 ICM   R14,B'1111',EXITRETN     GET RETURN ADDRESS
BR    R14  EXIT
LTORG
DROP
END
### Internal Field Values - UMSLXMAC

**Parameter-Related (directly)**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
<th>Macro Internal Field name</th>
<th>Value</th>
<th>Value</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASSMDT</td>
<td>YES</td>
<td>&amp;MDT</td>
<td>'1'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td>'0'</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>AUTORES</td>
<td>YES</td>
<td>&amp;RESET</td>
<td>'1'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td>'0'</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>OPTION1</td>
<td>LZFILL</td>
<td>&amp;OP1FLAG</td>
<td>'1'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTJUST</td>
<td>&amp;OP1FLAH</td>
<td>'1'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEVEL0</td>
<td>YES</td>
<td>&amp;WARN</td>
<td>'Y'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td>'N'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;null&gt;</td>
<td></td>
<td>'Y'</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>KEYFLD</td>
<td>0</td>
<td>&amp;KEYF</td>
<td>'000'</td>
<td></td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>'001'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>'011'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>'100'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The next three fields define the screen-field – main storage-field pair represented by this invocation of the macro. The main storage field is either GSAFLD or HOSTFLD, not both.

<table>
<thead>
<tr>
<th>MAPFLD</th>
<th>&lt;mapfldnm&gt;</th>
<th>&amp;MAPFLD</th>
<th>&lt;mapfldnm&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSAFLD</td>
<td>&lt;fname&gt;</td>
<td>&lt;null&gt;</td>
<td>&amp;INTFLD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;BASE</td>
<td>'UGCOMMON'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;OUTONL</td>
<td>'0'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;CHKGSA</td>
<td>'G$&amp;GSAFLD'</td>
</tr>
<tr>
<td>HOSTFLD</td>
<td>&lt;null&gt;</td>
<td>&lt;fname&gt;</td>
<td>&amp;INTFLD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;BASE</td>
<td>'UHCITEXT'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;OUTONL</td>
<td>'1'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;CHKGSA</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>VALEROR</td>
<td>&lt;err msg #&gt;</td>
<td>&lt;null&gt;</td>
<td>&amp;VALEROR</td>
</tr>
</tbody>
</table>
Sample Map Source Fragment

Note the use of the UMSHEADR macro to form the standard heading on the map (screen). See the assembled sample to view the generation of FLD0001 through FLD0004 and the (standard) header literals. See the sample map dsect (following) for the commarea definition. You may relate the use of the field names in the LX-Table macros for the generation of the ‘translate table.’

```
UGL0260  UMSHEADR LTH,'LICENSE TRANSACTION HISTORY'
          DFHMDF POS=(004,002),LENGTH=0001,INITIAL=' '
          DFHMDF POS=(005,001),LENGTH=0010,
                    INITIAL='LICENSE #' X
FLD0005  DFHMDF POS=(005,012),LENGTH=0025,
          JUSTIFY=(LEFT,BLANK), X
          ATTRB=(UNPROT,IC) X
          DFHMDF POS=(005,038),LENGTH=0006,
                    INITIAL='STATE:' X
FLD0006  DFHMDF POS=(005,045),LENGTH=0002,
          JUSTIFY=(LEFT,BLANK), X
          ATTRB=(UNPROT) X
          DFHMDF POS=(005,048),LENGTH=0004,
                    INITIAL='SS#:' X
FLD0007  DFHMDF POS=(005,053),LENGTH=0009,INITIAL=' ', X
          ATTRB=UNPROT
          DFHMDF POS=(005,063),LENGTH=0001
          DFHMDF POS=(006,004),LENGTH=0007,
                    INITIAL='NAME L:' X
FLD0010  DFHMDF POS=(006,012),LENGTH=0016,
          JUSTIFY=(LEFT,BLANK), X
          ATTRB=(UNPROT)
```
Sample Assembler Map Dsect

1111  COPY  UGL0260C
000FEA  006F8
1112  ORG  UGCUSERA
1113  * START OF GSA STORAGE
1114  LHLIC#  GFLD CL25   LH-IC NUMB
0006F8
1115+LHLIC#_Z  DS  CL1   ZFIELD
0006F9
1116+LHLIC#_T  DS  CL1   TFIELD
0006FA
1117+LHLIC#  DS  CL25   DATA FIELD

1118  LHSTATE  GFLD CL2   STATE
000713
1119+LHSTATE_Z  DS  CL1   ZFIELD
000714
1120+LHSTATE_T  DS  CL1   TFIELD
000715
1121+LHSTATE  DS  CL2   DATA FIELD

1122  LHSSN  GFLD CL9   SSN
000717
1123+LHSSN_Z  DS  CL1   ZFIELD
000718
1124+LHSSN_T  DS  CL1   TFIELD
000719
1125+LHSSN  DS  CL9   DATA FIELD
1126  LHLNAME  GFLD CL16   LAST-NAME
000722
1127+LHLNAME_Z  DS  CL1   ZFIELD
000723
1128+LHLNAME_T  DS  CL1   TFIELD
000724
1129+LHLNAME  DS  CL16   DATA FIELD
1130  LHFNNAME  GFLD CL12   FIRST-NAME
000734
1131+LHFNNAME_Z  DS  CL1   ZFIELD
000735
1132+LHFNNAME_T  DS  CL1   TFIELD

Data ommitted for brevity.

1209  LHLINE44  GFLD CL79   DETAIL LINE
00B7B
1210+LHLINE44_Z  DS  CL1   ZFIELD
00B7C
1211+LHLINE44_T  DS  CL1   TFIELD
Chapter Six - LXTABLE Processing

000B7D  1212+LHLINE44 DS   CL79    DATA FIELD
000BCC  1213 LHLINE45 GFLD CL79   DETAIL LINE
000BCE  1214+LHLINE45_Z DS   CL1    ZFIELD
000BCC  1215+LHLINE45_T DS   CL1    TFIELD
000BCE  1216+LHLINE45 DS   CL79    DATA FIELD
000C1D  1217 PHASE DS   XL1    PHASE COU
000C20  1218 PERSSURR DS   F      CURRENT P
000C24  1219 NUMOBREF DS   H      TOTAL BRE
000C26  1220 BREF1 DS   H      NUMBER OF
000C28  1221 BREFX DS   H      NUMBER OF
000C2A  1222 PAGEC DS   XL1    CURRENT P
000C2B  1223 PAGELIST DS   256XL1  FIRST REC
000C2D  1224 NOTFOUND DS   XL1    FLAG, NOT
      006FA 1225 LHGSA EQU LHLIC#,*-LHLIC#,C'X'GSA
000C2D  00FEA 1226 ORG
### LXTABLE Assembled Example

**EXTERNAL SYMBOL DICTIONARY**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>ID</th>
<th>ADDR</th>
<th>LENGTH</th>
<th>LD ID</th>
<th>FLAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXTABTST</td>
<td>SD</td>
<td>0001</td>
<td>000000</td>
<td>000117</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CSECT2</td>
<td>SD</td>
<td>0002</td>
<td>000118</td>
<td>00006C</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CSECT3</td>
<td>SD</td>
<td>0003</td>
<td>000188</td>
<td>00000D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>UGLTEST1</td>
<td>SD</td>
<td>0004</td>
<td>000198</td>
<td>000056</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

**LXTABTST - UMS SAMPLE LXTBL DEFINITION**

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE</th>
<th>STATEMENT</th>
<th>ASM</th>
<th>H</th>
<th>V</th>
<th>02</th>
<th>15.47</th>
<th>07/09/91</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>C&amp;C ASSOCIATES 1989</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>COPY ECELXMAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00110000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PUSH PRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00010000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PRINT OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00020000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>996</td>
<td>POP PRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>09700000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>998</td>
<td>UMSHEADR TYPE=DSECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00140000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>999+</td>
<td>MAPINP DSECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000+</td>
<td>HEADER OF INPUT-MAP/OUTPUT MAP AREA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001+</td>
<td>DS CL12 HEADER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1002+</td>
<td>DS CL3 ADDRESS/ATTRIBUTES FOR DATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1003+</td>
<td>MIDATE DS CL10 DATE TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1004+</td>
<td>DS CL3 ADDRESS/ATTRIBUTES FOR TIME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1005+</td>
<td>MITIME DS CL5 TIME TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1006+</td>
<td>DS CL3 ADDRESS/ATTRIBUTES FOR FUNC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1007+</td>
<td>MIFUNC DS CL4 FUNC TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1008+</td>
<td>DS CL3 ADDRESS/ATTRIBUTES FOR TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1009+</td>
<td>MITEXT DS CL50 MESG TEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010+</td>
<td>MI$SLEN EQU *-MAPINP BASE LENGTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01-UMSHE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011+</td>
<td>DS CL3 BMS DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00150000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1012+</td>
<td>MCHAR DS CL17 CHARACTER FIELD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00160000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1013+</td>
<td>DS CL3 BMS DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00170000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Length</td>
<td>Description</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>-------</td>
<td>--------</td>
<td>-------------------------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000074</td>
<td>MCHAR2</td>
<td>DS</td>
<td>CL10</td>
<td>CHARACTER FIELD</td>
<td>00180000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00007E</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00190000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000081</td>
<td>MLAST</td>
<td>DS</td>
<td>CL16</td>
<td>CHARACTER FIELD</td>
<td>00200000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000091</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00210000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000094</td>
<td>MFIRST</td>
<td>DS</td>
<td>CL12</td>
<td>CHARACTER FIELD</td>
<td>00220000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000099</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00230000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000A3</td>
<td>MMIDDLE</td>
<td>DS</td>
<td>CL8</td>
<td>CHARACTER FIELD</td>
<td>00240000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000AB</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00250000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000AE</td>
<td>MNUM1</td>
<td>DS</td>
<td>ZL5</td>
<td>NUMERIC FIELD</td>
<td>00260000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000B3</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00270000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000B6</td>
<td>MNUM2</td>
<td>DS</td>
<td>ZL7</td>
<td>NUMERIC FIELD</td>
<td>00280000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000BD</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00290000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000C0</td>
<td>MNUM3</td>
<td>DS</td>
<td>ZL4</td>
<td>NUMERIC FIELD</td>
<td>00300000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000C4</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00310000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000D1</td>
<td>MDATE</td>
<td>DS</td>
<td>CL10</td>
<td>DATE FIELD</td>
<td>00320000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000D4</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00330000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000DB</td>
<td>MDATE2</td>
<td>DS</td>
<td>CL7</td>
<td>DATE FIELD</td>
<td>00340000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000DE</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00350000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000E8</td>
<td>MDATEA</td>
<td>DS</td>
<td>CL10</td>
<td>DATE FIELD</td>
<td>00360000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000EB</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00370000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000F5</td>
<td>MDATEB</td>
<td>DS</td>
<td>CL10</td>
<td>DATE FIELD</td>
<td>00380000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000F8</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00390000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000F8</td>
<td>MPREF</td>
<td>DS</td>
<td>CL3</td>
<td>PREFIX</td>
<td>00400000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000FB</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00410000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000FE</td>
<td>MCOLOR</td>
<td>DS</td>
<td>CL1</td>
<td>COLOR</td>
<td>00420000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000FF</td>
<td>MREGNO</td>
<td>DS</td>
<td>CL7</td>
<td>REGNO</td>
<td>00430000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000102</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00440000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000109</td>
<td>MZIPC</td>
<td>DS</td>
<td>CL10</td>
<td>ZIPCODE</td>
<td>00450000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010C</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00460000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000116</td>
<td>MTITLE#</td>
<td>DS</td>
<td>CL8</td>
<td>TITLE#</td>
<td>00470000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000119</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00480000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000121</td>
<td>MLICN25</td>
<td>DS</td>
<td>CL25</td>
<td>LICNO</td>
<td>00490000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000124</td>
<td>DS CL3</td>
<td>BMS DATA</td>
<td>00500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00013D</td>
<td>MREGNO</td>
<td>DS</td>
<td>CL3</td>
<td>BMS DATA</td>
<td>00510000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter Six - LXTABLE Processing

000140 1048 MSTATE25 DS CL2 STATE 00520000

1050 COPY UGZCOMMA 00540000

1052 * UNINSURED MOTORIST. COMMON-AREA, GUEST SIDE. 00020000
1053 UGCOMMON DSECT 00030000
1054 UGCPROTO DS XL64 SYSTEM PROTOCALL DATA 00040000
1055 ORG UGCPROTO 00050000
1056 * WITH TWO EXCEPTIONS, THE PROTOCALL AREA IS
1057 * RESERVED FOR THE CONTROL
1058 * SOFTWARE. APPLICATIONS MAY USE THE
1059 * DUBLE-WORD TEMP AND SHOULD NOTE
1060 * THE QUALIFICATION ABOUT PREVIOUS MAPNAME
1061 UGCOMMON DS CL2 STATE 00060000
1064 UGCDTEMP DS D GENERAL PURPOSE DOUBLE-WORD TEMP 00090000
1065 UGCCLENG DS H LENGTH FOR USE WITH XCTL. 00100000
1066 UGCRDLEN DS H INQUIRY/REPLY DETAIL LENGTH 00110000
1067 UGCCURFN DS CL4 CURRENT FUNCTION-NAME 00120000
1068 UGCALLON DS XL1 FLAG BYTE ALL BITS INIT ON 00130000
1069 UGCMAPFI EQU B'00000001' MAP/DEMAP OPER. DEMAP 00140000
1070 UGCMAPFO EQU B'00000010' MAP/DEMAP OPER MAP 00150000
1071 UGCUSERA EQU * THE USER-PROGRAM AREA 00160000
1072 DS CL1 02030000
1073 DS 2290XL1 PUSH TO PAGE 02040000
1074 DS 2290XL1 02050000
1075 TOTAL LENGTH 02060000
1076 THE LENGTH OF THIS AREA IS TAKEN FROM
1077 * THE PROCESS-CONTROL TABLE 02070000
1078 BUT WILL NOT BE LESS THAN THE VALUE OF
1079 * THE ABOVE EQUATE AT THE LAST 02080000
1080 ASSEMBLY OF UGZ0015P 02090000
1081 ORG UGCUSERA 00050000
1082 ASSEMBLY OF UGZ0015P 00550000
1083 ORG UGCUSERA 00560000
1084 GTITLE# DS CL8 TITLE# 00570000
1085 GTITLE# DS CL8 TITLE# 00580000
## Chapter Six - LXTABLE Processing

<table>
<thead>
<tr>
<th>Line</th>
<th>Symbol</th>
<th>Type</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>000702</td>
<td>1282</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00590000</td>
</tr>
<tr>
<td>000704</td>
<td>1283</td>
<td>GDATE</td>
<td>DS HL2 BINARY-DATE (MUST BE H OR HL2)</td>
<td>00600000</td>
</tr>
<tr>
<td>000706</td>
<td>1284</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00610000</td>
</tr>
<tr>
<td>000708</td>
<td>1285</td>
<td>GDATE2</td>
<td>DS HL2 BINARY-DATE (MUST BE H OR HL2)</td>
<td>00620000</td>
</tr>
<tr>
<td>00070A</td>
<td>1286</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00630000</td>
</tr>
<tr>
<td>00070C</td>
<td>1287</td>
<td>GDATEA</td>
<td>DS HL2 BINARY-DATE (MUST BE H OR HL2)</td>
<td>00640000</td>
</tr>
<tr>
<td>00070E</td>
<td>1289</td>
<td>GDATEB</td>
<td>GFLD HL2</td>
<td>00660007</td>
</tr>
<tr>
<td>00070F</td>
<td>1290+</td>
<td>GDATEB_Z</td>
<td>DS CL1 ZFIELD</td>
<td>01-00873</td>
</tr>
<tr>
<td>000710</td>
<td>1291+</td>
<td>GDATEB_T</td>
<td>DS CL1 TFIELD</td>
<td>01-00874</td>
</tr>
<tr>
<td>000712</td>
<td>1292+</td>
<td>GNUMB</td>
<td>DS HL2 DATA FIELD</td>
<td>01-00875</td>
</tr>
<tr>
<td>000713</td>
<td>1294</td>
<td>GNUMB</td>
<td>GFLD FL4</td>
<td>00680007</td>
</tr>
<tr>
<td>000714</td>
<td>1295+</td>
<td>GNUMB_Z</td>
<td>DS CL1 ZFIELD</td>
<td>01-00873</td>
</tr>
<tr>
<td>000715</td>
<td>1296+</td>
<td>GNUMB_T</td>
<td>DS CL1 TFIELD</td>
<td>01-00874</td>
</tr>
<tr>
<td>000716</td>
<td>1297+</td>
<td>GNUMB</td>
<td>DS FL4 DATA FIELD</td>
<td>01-00875</td>
</tr>
<tr>
<td>000718</td>
<td>1299 *</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00700007</td>
</tr>
<tr>
<td>000719</td>
<td>1300 *</td>
<td>GDATEB</td>
<td>DS HL2 BINARY-DATE (MUST BE H, HL2)</td>
<td>00710007</td>
</tr>
<tr>
<td>00071A</td>
<td>1301 *</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00720007</td>
</tr>
<tr>
<td>00071B</td>
<td>1302 *</td>
<td>GNUMB</td>
<td>DS FL4 BINARY ITEM (MUST BE F, FL4)</td>
<td>00730007</td>
</tr>
<tr>
<td>00071C</td>
<td>1304</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00750000</td>
</tr>
<tr>
<td>00071D</td>
<td>1305</td>
<td>GNUMP</td>
<td>DS PL5 PACKED FIELD (WATCH LENGTH)</td>
<td>00760000</td>
</tr>
<tr>
<td>00071E</td>
<td>1306</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00770000</td>
</tr>
<tr>
<td>00071F</td>
<td>1307</td>
<td>GNUMZ</td>
<td>DS ZL5 NUMERIC (MUST = LEN OF SOURCE)</td>
<td>00780000</td>
</tr>
<tr>
<td>000720</td>
<td>1308</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00790000</td>
</tr>
<tr>
<td>000721</td>
<td>1309</td>
<td>GCHAR</td>
<td>DS CL17 CHAR. (MUST = LEN OF SOURCE)</td>
<td>00800000</td>
</tr>
<tr>
<td>000722</td>
<td>1310</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00810000</td>
</tr>
<tr>
<td>000723</td>
<td>1311</td>
<td>GLAST</td>
<td>DS CL16 CHAR. (MUST = LEN OF SOURCE)</td>
<td>00820000</td>
</tr>
<tr>
<td>000724</td>
<td>1312</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00830000</td>
</tr>
<tr>
<td>000725</td>
<td>1313</td>
<td>GFIRST</td>
<td>DS CL12 CHAR. (MUST = LEN OF SOURCE)</td>
<td>00840000</td>
</tr>
<tr>
<td>000726</td>
<td>1314</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00850000</td>
</tr>
<tr>
<td>000727</td>
<td>1315</td>
<td>GMIDDLE</td>
<td>DS CL8 CHAR. (MUST = LEN AS SOURCE)</td>
<td>00860000</td>
</tr>
<tr>
<td>000728</td>
<td>1316</td>
<td>DS</td>
<td>CL1,CL1 ZFLD, TFLD</td>
<td>00870000</td>
</tr>
<tr>
<td>000729</td>
<td>1317</td>
<td>GPREF</td>
<td>DS CL3 CHAR. (MUST = LEN AS SOURCE)</td>
<td>00880000</td>
</tr>
</tbody>
</table>
**Chapter Six - LXTABLE Processing**

```
000768  1318   DS   CL1,CL1       ZFLD, TFLD             0089000
00076A  1319   GCOLOR  DS   CL1      CHAR.(MUST = LEN AS SOURCE) 0090000
00076B  1320   DS   CL1,CL1       ZFLD, TFLD             0091000
00076D  1321   GREGNO  DS   CL7      CHAR.(MUST = LEN AS SOURCE) 0092000
000774  1322   DS   CL1,CL1       ZFLD, TFLD             0093000
000776  1323   GZIPC   DS   XL6      CHAR.(MUST = LEN AS SOURCE) 0094000
00077C  1324   DS   CL1,CL1       ZFLD, TFLD             0095000
00077E  1325   GICODE1 DS   CL30     CHAR.(MUST = LEN AS SOURCE) 0096000
00079C  1326   GICODE2 DS   CL18     CHAR.(MUST = LEN AS SOURCE) 0097000
0007B0  1327   GXCODEA DS   PL2      CHAR.(MUST = LEN AS SOURCE) 0100000
0007B4  1328   GSTATE25 DS    CL2     CHAR.(MUST = LEN AS SOURCE) 0101000
0007B6  1329   GLICN25 DS    CL25    CHAR.(MUST = LEN AS SOURCE) 0102000
0007B8  1332   DS   CL1,CL1       ZFLD, TFLD             0103000
0007BA  1333   GLICN25 DS    CL25    CHAR.(MUST = LEN AS SOURCE) 0104000
0007BB  1335   COPY   UHZCOMMA

00076D  1337   ***** UMS COMMON-AREA, HOST SIDE. ** ECE *** 00020000
000000  1338   UHCOMMON DSECT
000000  1339   UHCPROTO DS    XL64     SYSTEM PROTOCALL DATA 00040000
000040  00000  1340   ORG   UHCPROTO 00050000
000000  1341   UHCDTEMP DS    D       GENERAL PURPOSE DBL-WORD TEMP. 00060000
000000  1342   UHCPARM1 EQU   UHCDTEMP,4 SPECIAL-PURPOSE PARM-1 00070000
000000  1343   UHCPARM2 EQU   UHCDTEMP+4,4 SPECIAL-PURPOSE PARM-2 00080000
000000  1344   UHCCLENG DS    H       LENGTH FOR USE WITH XCTL. 00090000
000000  1345   UHCRDLEN DS    H       RESPONSE DETAIL LENGTH 00100000
000000  1346   UHCEFLAG DS    XL1      ENTRY-REASON.
                      X'FF'=DUPKEY RESOLUTION 00110000
000000  1347   DS   XL1      FILLER 00120000
00000E  1348   UHCDUPNM DS    CL4     HOLD DUPKEY NAME BY DISPATCHER 00130000
000012  1349   UFHPINS2 DS    10PL2    ADDED INSURANCE-CODE LIST 00140000
000026  1350   UHPCTIND DS    H       HOST PCT INDEX. X'FFFF' = NONE 00150000
000028  1351   UHSTARTT DS    D       R/T CLOCK AT UH20001P INITI. 00160000
```

Chapter Six - LXTABLE Processing
1492+ DC BL1'00000000' FLAGS 01-00118
1493+ DC XL3'0' RESERVED 01-00119
1494+ DC XL2'FFFF' TEMP TRAILER 01-00120
1496 * 01990008
1497 * EXAMPLE: CHARACTER TO CHARACTER, FIELDS MUST BE SAME LENGTH
1499 UMSLXTBL MAPFILD=MCHAR,GSFILD=GCHAR,EDIT=VIN,KEYFILD=1
000027 00025 1500+ ORG *-2 01-00345
1501+G$GCHAR DC AL1(L0005-*) , AL1(008) 01-00346
1502+M$MCHAR DC BL1'00100000' 01-00347
1503+ DC BL1'00010' 01-00348
1504+ DC AL2 (GCHAR-UGCOMMON-2) 01-00349
1505+ DC AL2 (MCHAR-MAPINP-3) 01-00350
1506+ DC AL1 (L'GCHAR-1), AL1 (L'MCHAR-1) 01-00351
1507+E$GCHAR DS 0AL1 DEFINE REFERENCE 02-00531
1508+ DC AL4 (E0006) POINT TO EDIT ARRAY 02-00535
1509+CSECT2 CSECT 02-00536
1510+E0006 DC AL1 (L0006-*) LENGTH OF ARRAY 02-00537
1511+ DC AL1 (00), AL2 (1) MODE FLAG BITS / TYPE 02-00755
1512+ DC AL4 (G$GCHAR) REFERENCE TO LXTBL BLOCK 02-00770
1513+L0006 DS 0AL1 END OF ARRAY 02-00771
1514+LXTABTST CSECT 02-00771
1515+L0005 DC XL2 'FFFF' 01-00361
1517 * 01240008
1518 * EXAMPLE: CHARACTER TO CHARACTER,
* FIELDS MUST BE SAME LENGTH, OUTPUT 01250008
1520 UMSLXSTBL MAPFILD=MCHAR2,HOSTFILD=HCHAR 01270008
00035 00033 1521+ ORG *-2 01-00345
1522+ DC AL1 (L0007-*) , AL1 (008) 01-00346
1523+M$MCHAR2 DC BL1'00001000' 01-00347
1524+ DC BL1'00000' 01-00348
1525+ DC AL2 (HCHAR-UHCITEXT-2) 01-00349
1526+ DC AL2 (MCHAR2-MAPINP-3) 01-00350
1527+ DC AL1 (L'HCHAR-1), AL1 (L'MCHAR2-1) 01-00351
1528+L0007 DC XL2'FFFF' 01-00361
1530 * 01290008
1531 * EXAMPLE: NUMERIC TO NUMERIC, FIELDS MUST BE SAME LENGTH
1533 UMSLXTBL MAPFLD=MNUM1,GSAFLD=GNUM2,
VALUE=(RALLOW,1,2,3,4,5670) 01320008
1534+ ORG *-2 01-00345
00003F 0003D
1535+G$GNUMZ DC AL1(L0008-*),AL1(000) 01-00346
000040 00
1536+M$MNUM1 DC BL1'00010001' 01-00347
1537+ DC BL1'00000' 01-00348
1538+ DC AL2(GNUMZ-UGCOMMON-2) 01-00349
1539+ DC AL2(MNUM1-MAPINP-3) 01-00350
1540+ DC AL1(L'GNUMZ-1),AL1(L'MNUM1-1) 01-00351
1541+ DC AL4(V0009) POINT TO VALUE LIST 02-00431
1542+CSECT2 CSECT 02-00432
1543+V0009 DC BL1'0001' FLAGS 02-00436
1544+ DC AL1(3) TRUE ENTRY LENGTH 02-00437
1545+ DC AL2(4) NUMBER OF LIST ELEMENTS 02-00438
1546+ DC PL3'1' 02-00473
1547+ DC PL3'2' 02-00473
1548+ DC PL3'3' 02-00473
1549+ DC PL3'45670' 02-00473
1550+LXNTABST CSECT 02-00478
1551+L0008 DC XL2'FFFF' 01-00361
1552+L0008
1553 * 01340008
1554 * EXAMPLE: NUMERIC TO PACKED, OUTPUT MUST BE LONG-ENOUGH
1556 UMSLXTBL MAPFLD=MNUM2,GSAFLD=GNUMP,
OPTION1=LZFILL,EXIT=USER01,KEYFLD=4 X
1557+EXITDATA DSECT 01370008
000000
1558+EXITCODE DS AL2 UNIQUE IDENTIFIER CODE SC1290 02-00910
1559+EXITWHYS DS XL1 SUMMATION OF ENTRY REASONS 02-00911
1560+EXITRETN DS AL4 RETURN ADDRESS SC1290 02-00912
1561+EXITWHYX DS XL1 REASON FOR THIS ENTRY SC1290 02-00913
00001 1562+EXRVALOK EQU B'000000001' VALUE IS OK SC1290 02-00914
00002 1563+EXRVALNG EQU B'00000010' VALUE IS NOT OK SC1290 02-00915
00004  1564+EXREDTOK EQU B'00000100' EDIT IS OK SC1290 02-00916
00008  1565+EXREDTNG EQU B'00000100' EDIT IS NOT OK SC1290 02-00917
00010  1566+EXRALWAY EQU B'00010000' ALWAYS ENTER UNLESS FMT ERR02-00918
00008  1567+EXITUSER DS 0XL5 USER PARMs SC1290 02-00919
00008  1568+EXITACTN DS XL1 USER ACTION REQUEST SC1290 02-00920
00000  1569+EXANOACT EQU B'00000000' CONTINUE, NO ACTION SC1290 02-00921
00001  1570+EXANOERR EQU B'00000001' CONTINUE, NO ERROR SC1290 02-00922
00009  1571+EXAWIERR EQU B'00000010' CONTINUE, POST ERR. SC1290 02-00923
00009  1572+EXITEROR DS XL4 USER ERROR CODE SC1290 02-00924
0004D  1573+LXTABTST CSECT
1574+UMSLXTBL/EXIT REASON DEFAULTED TO ALL (FF) 02-00925
1575+UMSLXTBL/EXIT LOCATION-CODE DEFAULTED TO 65535 02-00965
0004D  1576+ ORG *-2 01-00345
0004B 1104  1577+G$GNUMP DC AL1(L0010-*),AL1(004) 01-00346
0004D 81  1578+M$MNUM2 DC BL1'10000001' 01-00347
0004E 18  1579+ DC BL1'11000' 01-00348
00004F 0718  1580+ DC AL2(00000001' 01-00349
000051 00B3  1581+ DC AL2(MNUM2-MAPINP-3) 01-00350
000053 0406  1582+ DC AL1(L0012-1),AL1(L0012-1) 01-00351
000055 00000198FFFFFF + EXIT STRING VALUE
1583+ DC AL4(USER01),XL1'FF',AL2( 5553) X01-00359
0005C FFFF  EXIT STRING VALUE
1584+L0010 DC XL2'FFFF' 01-00361
1586 * 01390008
1587 * EXAMPLE: NUMERIC TO BINARY, I/P MUST BE 1-9 DIGITS 01400008
1588 * OUTPUT MUST BE F OR FL4 01410000
1589 UMSLXTBL MAPFLD=MNUM3,GSAFLD=GNUMB,OPTION1=RTJUST,
KEYFLD=1
0005C 0A0C  1590+ ORG *-2 01-00345
0004E 01  1591+G$GNUMB DC AL1(L0012-*),AL1(012) 01-00346
00005F 03  1592+M$MNUM3 DC BL1'00000001' 01-00347
000060 0712  1593+ DC BL1'000011' 01-00348
00062 00BD  1594+ DC AL2(00000001' 01-00349
000064 0303  1595+ DC AL2(MNUM3-MAPINP-3) 01-00350
000064 0303  1596+ DC AL1(L0012-1),AL1(L0012-1) 01-00351

Chapter Six - LXTABLE Processing
1597+L0012 DC XL2'FFFF' 01-00361
1599 * 01440008
1600 * EXAMPLE: DATE, INPUT MUST BE CL10, OUTPUT MUST BE H OR HL2
1602 UMSLXTBL MAPFLD=MDATE, GSAFLD=GDATE, EDIT=AGE1600,
VALUE=(ALLOW, 8000, 0000, FFFF, 7FFF),
EXIT=(USER02, 01, 99) 01480008

000068                      00066  1603+         ORG   *-2                                    01-00345
000066 1910                        1604+G$GDATE  DC    AL1(L0013-*) , AL1(016)                  01-00346
000068 32                          1605+M$MDATE  DC    BL1'00110010'                          01-00347
000069 10                          1606+         DC    BL1'10000'                             01-00348
00006A 0702                        1607+         DC    AL2 (GDATE-UGCOMMON-2)                    01-00349
00006C 00C4                        1608+         DC    AL2 (MDATE-MAPINP-3)                     01-00350
00006E 0109                        1609+         DC    AL1 (L'GDATE-1), AL1 (L'MDATE-1)               01-00351
000070 00000130                    1610+         DC    AL4 (V0015)      POINT TO VALUE LIST    02-00431
000130 1611+CSECT2 CSECT
000130 04                          1612+V0015    DC    BL1'0100'       FLAGS                  02-00436
000131 02                          1613+         DC    AL1(2)          TRUE ENTRY LENGTH      02-00437
000132 0004                        1614+         DC    AL2 (4)        NUMBER OF LIST ELEMENTS02-00438
000134 0000                        1615+         DC    AL2 (X'8000'+32768)                     02-00473
000136 8000                        1616+         DC    AL2 (X'0000'+32768)                     02-00473
000138 7FFF                        1617+         DC    AL2 (X'FFFF'+32768)                     02-00473
00013A FFFF                      1618+         DC    AL2 (X'7FFF'+32768)                     02-00473
000074 1619+LXTABTST CSECT
000074 1620+E$GDATE DS 0AL1 DEFINE REFERENCE 02-00531
000074 0000013C                    1621+         DC    AL4 (E0016)      POINT TO EDIT ARRAY    02-00535
00013C 1622+CSECT2 CSECT
00013C 08                          1623+E0016    DC    AL1 (L0016-*)    LENGTH OF ARRAY 02-00536
00013D 000012                        1624+         DC    AL1 (00), AL2 (18) MODE FLAG BITS / TYPE 02-00755
000140 00000066                    1625+         DC    AL4 (G$GDATE)  REFERENCE TO LXTBL BLK 02-00766
000144 1626+L0016 DS 0AL1 END OF ARRAY 02-00770
000078 1627+LXTABTST CSECT
000078 000001B8010063 1628+         DC    AL4 (USER02), XL1'01', AL2 (99) X01-00359
+ EXIT STRING VALUE
00007F FFFF                      1629+L0013 DC XL2'FFFF' 01-00361
1631 * EXAMPLE: EXPDATE, INPUT MUST BE CL7,  
* OUTPUT MUST BE H OR HL2
1633 UMSLXTBL MAPFLD=MDATE2,GSAFLD=GDAT2E 01520008
1635+ ORG *-2 01-00345
000081 0007F 1634+ DC AL1(L0017-*),AL1(020) 01-00346
000082 04 1636+ DC BL1'000001000' 01-00347
000083 00 1637+ DC BL1'00000' 01-00348
000085 00D1 1638+ DC AL2(GDATE2-UGCOMMON-2) 01-00349
000087 0106 1639+ DC AL2(MDATE2-MAPINP-3) 01-00350
000089 FFFF 1638+ DC AL1(L'GDATE2-1),AL1(L'MDATE2-1) 01-00351
1640+ DC AL1(L'GDATE2-1),AL1(L'MDATE2-1) 01-00351
1641+L0017 DC XL2'FFFF' 01-00361

1643 * EXAMPLE: COMPLETE SET FOR REGISTRATION 01540008
1645 UMSLXTBL MAPFLD=MPREF,GSAFLD=GPREF 01560008
1646+ ORG *-2 01-00345
00008B 00089 1647+ DC AL1(L0018-*),AL1(008) 01-00346
00008B 00 1648+ DC BL1'000000000' 01-00347
00008C 00 1649+ DC BL1'00000' 01-00348
00008D 0763 1650+ DC AL2(GPREF-UGCOMMON-2) 01-00349
00008F 00F5 1651+ DC AL2(MPREF-MAPINP-3) 01-00350
000091 0202 1652+ DC AL1(L'GPREF-1),AL1(L'MPREF-1) 01-00351
000093 FFFF 1653+L0018 DC XL2'FFFF' 01-00361
1655 UMSLXTBL MAPFLD=MCLR,GSAFLD=GCLR 01580008
1656+ ORG *-2 01-00345
000095 00093 1657+ DC AL1(L0019-*),AL1(008) 01-00346
000095 00 1658+ DC BL1'000000000' 01-00347
1658+ DC BL1'00000' 01-00348
000096 00 1659+ DC AL2(GCOLOR-UGCOMMON-2) 01-00349
000098 00FB 1660+ DC AL2(MCOLOR-MAPINP-3) 01-00350
00009B 0000 1661+ DC AL1(L'GCOLOR-1),AL1(L'MCOLOR-1) 01-00351
00009D FFFF 1662+ DC AL1(L'GCOLOR-1),AL1(L'MCOLOR-1) 01-00351
1663+L0020 DC XL2'FFFF' 01-00361

Chapter Six - LXTABLE Processing

1665 UMSLXTBL MAPFLD=MDATEA, GSAFLD=GDATEA 01600008
00009F 0009D 1666+ ORG *-2 01-00345
00009D 0A10 1667+G$GDATEA DC AL1(L0020-*), AL1(016) 01-00346
00009F 02 1668+M$MDATEA DC BL1'00000010' 01-00347
0000A0 00 1669+ DC BL1'000000' 01-00348
0000A1 070A 1670+ DC AL2(GDATEA-UGCOMMON-2) 01-00349
0000A3 00D8 1671+ DC AL2(MDATEA-MAPINP-3) 01-00350
0000A5 0109 1672+ DC AL1(L'GDATEA-1), AL1(L'MDATEA-1) 01-00351
0000A7 FFFF 1673+L0020 DC XL2'FFFF' 01-00361

1675 UMSLXTBL MAPFLD=MDATEB, GSAFLD=GDATEB, X01620008
EDIT=(PDATE,GDATEA,GDATEB) 01630008
0000A9 000A7 1676+ ORG *-2 01-00345
0000A7 0E10 1677+G$GDATEB DC AL1(L0021-*), AL1(016) 01-00346
0000A9 22 1678+M$MDATEB DC BL1'00100010' 01-00347
0000AA 00 1679+ DC BL1'000000' 01-00348
0000AB 070E 1680+ DC AL2(GDATEB-UGCOMMON-2) 01-00349
0000AD 00E8 1681+ DC AL2(MDATEB-MAPINP-3) 01-00350
0000AF 0109 1682+ DC AL1(L'GDATEB-1), AL1(L'MDATEB-1) 01-00351
0000B1 1683+E$GDATEA DS 0AL1 DEFINE USAGE 02-00518
0000B1 1684+E$GDATEB DS 0AL1 DEFINE USAGE 02-00518
0000B1 0000144 1685+ DC AL4(E0022) POINT TO EDIT ARRAY 02-00535
000144 1686+CSECT2 CSECT 02-00536
000144 0C 1687+E0022 DC AL1(L0022-*) LENGTH OF ARRAY 02-00537
000145 000009 1688+ DC AL1(00), AL2(9) MODE FLAG BITS / TYPE 02-00755
000148 0000009D 1689+ DC AL4(G$GDATEA) REFERENCE TO LXTBL BLK 02-00762
00014C 000000A7 1690+ DC AL4(G$GDATEB) REFERENCE TO LXTBL BLK 02-00762
000150 1691+L0022 DS 0AL1 END OF ARRAY 02-00770
0000B5 1692+LXTABTST CSECT 02-00771
0000B5 FFFF 1693+L0021 DC XL2'FFFF' 01-00361

1695 UMSLXTBL MAPFLD=MLAST, GSAFLD=GLAST 01650008
0000B7 000B5 1696+ ORG *-2 01-00345
0000B5 0A08 1697+G$GLAST DC AL1(L0023-*), AL1(008) 01-00346
Chapter Six - LXTABLE Processing
Chapter Six - LXTABLE Processing


00015C 1731+L0026 DS OAL1 END OF ARRAY 02-00770
0000D7 1732+LXTABLE CSECT 02-00771
0000D7 FFFF 1733+L0025 DC XL2'FFFF' 01-00361

1735 UMSLXTBL MAPFLD=MFIRST, GSAFLD=GFIRST 01720008

0000D9 000D7 1736+ ORG *-2 01-00345
0000D7 0A08 1737+GS$FIRST DC AL1(L0027-*) AL1(008) 01-00346
0000D9 00 1738+MS$FIRST DC BL1'00000000' 01-00347
0000DA 00 1739+ DC BL1'00000' 01-00348
0000DB 074B 1740+ DC AL2(GFIRST-UGCOMMON-2) 01-00349
0000DD 0091 1741+ DC AL2(MFIRST-MAPINP-3) 01-00350
0000DF 0B0B 1742+ DC AL1(L'GFIRST-1), AL1(L'MFIRST-1) 01-00351
0000E1 FFFF 1743+L0027 DC XL2'FFFF' 01-00361

1745 UMSLXTBL MAPFLD=MMIDDLE, GSAFLD=GMIDDLE, EDIT=(PNAME, GLAST, GFIRST, GMIDDLE) X01740008

0000E3 000E1 1746+ ORG *-2 01-00345
0000E1 0E08 1747+GS$MIDDLE DC AL1(L0027-*) AL1(008) 01-00346
0000E3 20 1748+MS$MIDDLE DC BL1'00000000' 01-00347
0000E4 00 1749+ DC BL1'00000' 01-00348
0000E5 0759 1750+ DC AL2(GMIDDLE-UGCOMMON-2) 01-00349
0000E7 00A0 1751+ DC AL2(MMIDDLE-MAPINP-3) 01-00350
0000E9 0707 1752+ DC AL1(L'GMIDDLE-1), AL1(L'MMIDDLE-1) 01-00351
0000EB 1753+ES$GLAST DS OAL1 DEFINE USAGE 02-00518
0000EB 1754+ES$FIRST DS OAL1 DEFINE USAGE 02-00518
0000EB 1755+ES$MIDDLE DS OAL1 DEFINE USAGE 02-00518
0000EB 000015C 1756+ DC AL4(E0029) POINT TO EDIT ARRAY 02-00535
00015C 1757+CSECT2 CSECT 02-00536
00015C 10 1758+E0029 DC AL1(L0029-*) LENGTH OF ARRAY 02-00537
00015D 0000A 1759+ DC AL1(00), AL2(10) MODE FLAG BITS / TYPE 02-00755
000160 00000B5 1760+ DC AL4(GS$GLAST) REFERENCE TO LXTABLE BLK 02-00762
000164 00000D7 1761+ DC AL4(GS$FIRST) REFERENCE TO LXTABLE BLK 02-00762
000168 00000E1 1762+ DC AL4(GS$MIDDLE) REFERENCE TO LXTABLE BLK 02-00762
00016C 1763+L0029 DS OAL1 END OF ARRAY 02-00770
Chapter Six - LXTABLE Processing


0000EF 1764+LXTABTST CSECT 02-00771
0000EF FFFF 1765+L0028 DC XL2'FFFF' 01-00361

1767 UMSLXTBL MAPFLD=MREGNO,GSAFLD=GREGNO, EDIT=(REGNO,GPREF,GCOLOR,GREGNO) X01770008
               ORG *-2 01780008
0000F1 000EF 1768+ ORG *-2 01-00345
0000EF 0E08 1769+G$GREGNO DC AL1(L0030-*),AL1(008) 01-00346
0000F1 20 1770+M$MREGNO DC BL1'00100000' 01-00347
0000F2 00 1771+ DC BL1'00000' 01-00348
0000F3 076B 1772+ DC AL2(GREGNO-UGCOMMON-2) 01-00349
0000F5 00FF 1773+ DC AL2(MREGNO-MAPINP-3) 01-00350
0000F7 0606 1774+ DC AL1(L'GREGNO-1),AL1(L'MREGNO-1) 01-00351
0000F9 0000016C 1778+ DC AL4(E0031) POINT TO EDIT ARRAY 02-00535
00016C 1779+CSECT2 CSECT 02-00536
00016D 10 1780+E0031 DC AL1(L0031-*) LENGTH OF ARRAY 02-00537
000170 0000004 1781+ DC AL1(00),AL2(4) MODE FLAG BITS / TYPE 02-00755
000171 00000089 1782+ DC AL4(G$GPREF) REFERENCE TO LXTBL BLK 02-00762
000174 00000093 1783+ DC AL4(G$GCOLOR) REFERENCE TO LXTBL BLK 02-00762
000178 000000EF 1784+ DC AL4(G$GREGNO) REFERENCE TO LXTBL BLK 02-00762
00017C 1785+L0031 DS OAL1 END OF ARRAY 02-00770
0000FD 1786+LXTABTST CSECT 02-00771
0000FD FFFF 1787+L0030 DC XL2'FFFF' 01-00361
0000FD 1789 UMSLXTBL MAPFLD=MZIPC,GSAFLD=GZIPC 01800008
0000FF 000FD 1790+ ORG *-2 01-00345
0000FD 0A18 1791+G$GZIPC DC AL1(L0032-*) AL1(024) 01-00346
0000FF 40 1792+M$MZIPC DC BL1'01000000' 01-00347
000100 00 1793+ DC BL1'00000' 01-00348
000101 0774 1794+ DC AL2(GZIPC-UGCOMMON-2) 01-00349
000103 0109 1795+ DC AL2(MZIPC-MAPINP-3) 01-00350
000105 0509 1796+ DC AL1(L'GZIPC-1),AL1(L'MZIPC-1) 01-00351
000107 FFFF 1797+L0032 DC XL2'FFFF' 01-00361
Chapter Six - LXTABLE Processing

1799    UMSLXCON GSAINPUT=GXCODEA, GSAOUTPUT=GICODE1, TYPE=ICODE
000109   0001D 1800+ ORG LXTABTST+X'1D'
00001D  00000188
000021   1801+ DC AL4(CONTABLE) CONVERSION TABLE PTR
000109   01802+ ORG
000188   01803+CSECT3 CSECT
00001D   01804+ DC BL1'00000001' SET END LAG
000109   01805+ LXTABTST CSECT
000189   01806+CSECT3 CSECT
00001D   01807+ ORG '*-1
000188   00188 0000 DC BL1'00', AL1(0)
00001D   01808+ DC BL1'00', AL1(0)
00018A  079B   01809+S#GXCODEA DC AL2(GXCODEA-UGCOMMON-2)
00018C  077C   01810+ DC AL2(GICODE1-UGCOMMON-2)
00001D   01811+ T#GICODE1 DC BL1'00000001'
000189   01812+ LXTABTST CSECT
00001D   01814    UMSLXCON HOSTINPUT=HXCODEB, GSAOUTPUT=GICODE2, TYPE=ICODE
00018F   0018E 01815+CSECT3 CSECT
00018F   0018E 1816+ ORG '*-1
00001D   01817+ DC BL1'10', AL1(1)
00018E  0201   01818+S#HXCODEB DC AL2(HXCODEB-UHCITEXT-2)
000190  0046   01819+ DC AL2(GICODE2-UGCOMMON-2)
000192  079C   01820+ T#GICODE2 DC BL1'00000001'
00001D   01821+ LXTABTST CSECT
00001D   01823    UMSLXTBL MAPFLD=MTITLE#, GSAFLD=GTITLE#, EDIT=TITLE#
00001D   00107 1824+ ORG '*-2
00001D   00107 1825+ G$GTITLE# DC AL1(L0035-*), AL1(008)
00001D   00108 1826+ M$MTITLE# DC BL1'00100000'
00001D   00109 1827+ DC BL1'000000'
00001D   00109 1828+ DC AL2(GTITLE#-UGCOMMON-2)
00001D   00109 1829+ DC AL2(MTITLE#-MAPINP-3)
00001D   00109 1830+ DC AL1(L'GTITLE#-1), AL1(L'MTITLE#-1)
00001D   00111 1831+E$GTITLE# DS 0AL1 DEFINE REFERENCE
Chapter Six - LXTABLE Processing

```
000111 0000017C 00017C 00017C 08 00017D 000008 000180 00000107 000184 000115 000115 FFFF
000198 00000 00000 00000 00000 00000 000198 00198 00198 D603 96C0 96C0 006C0 006C0
00019E 4780 C016 001AE 4780 C016 001AE 9201 8008 00008 0019E 1858 USER01,DS,0H
0001A2 910A 8007 00007 0019E 1859 OC,UGGERRCD,UGGERRCD,PREVIOUS ERROR?,02060000
0001A6 4780 C016 001AE 1861 TM,EXITWHYX,EXRVALNG+EXREDTNG,ERROR?,02080000
0001AA 9201 8008 00008 001A6 1862 BZ,EXITACTN,EXANOERR:YES,FORCE ERROR,02100000
0001AE 1865 U19999,DS,0H,02120000
0001AE BFEF 8003 00003 00198 00198 1857 USING USER01,R12,RUTINE BASE,02040000
000198 1858 USER01,DS,0H,02050000
00019E 4780 C016 001AE 1860 BZ,U19999,NO,02070000
0001A2 910A 8007 00007 0019E 1859 OC,UGGERRCD,UGGERRCD,PREVIOUS ERROR?,02060000
0001A6 4780 C016 001AE 1862 BZ,U19999,NO,02090000
0001AA 9201 8008 00008 0019E 1858 USER01,DS,0H,02050000
000198 D603 96C0 96C0 006C0 006C0 00019E 4780 C016 001AE 1860 BZ,U19999,NO,02070000
0001A2 910A 8007 00007 0019E 1859 OC,UGGERRCD,UGGERRCD,PREVIOUS ERROR?,02060000
0001A6 4780 C016 001AE 1862 BZ,U19999,NO,02090000
0001AA 9201 8008 00008 0019E 1858 USER01,DS,0H,02050000
0001AE BFEF 8003 00003 00198 00198 1857 USING USER01,R12,RUTINE BASE,02040000
```

**USER EXITS: ALL GO HERE, AT THE END OF THE LXTABLE**

**NOTE: LEADING 'DROP' IS VERY IMPORTANT**

**THIS EXAMPLE WILL FORCE NO-ERROR IF THERE IS AN EXISTING ERROR AND WE HAVE BEEN ENTERED FOR THE REASON OF AN ERROR. IN ANY OTHER CASE, NORMAL FLOW WILL BE OBSERVED**

---

**Chapter Six - LXTABLE Processing**

133

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM</th>
<th>H</th>
<th>V</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001B2</td>
<td>07FE</td>
<td></td>
<td></td>
<td>1867</td>
<td>BR R14 EXIT</td>
<td>02140000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001B8</td>
<td></td>
<td></td>
<td></td>
<td>1869</td>
<td>LTORG</td>
<td>02160000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1871</td>
<td>DROP</td>
<td>02180000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>1872</td>
<td>USING UGCOMMON,R9</td>
<td>COMMON AREA</td>
<td>02190000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000</td>
<td>1873</td>
<td>USING EXITDATA,R8</td>
<td>EXIT DATA</td>
<td>02200000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

************************************************************************************
1875 * THIS EXAMPLE WILL FORCE AN ERROR IF  **
* GDATE = X'7FFF' AND GCHAR = ALL '$'  ** 02210000
1876 * IN ANY OTHER CASE, NORMAL FLOW WILL BE OBSERVED  ** 02220000
************************************************************************************

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM</th>
<th>H</th>
<th>V</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>001B8</td>
<td>1878</td>
<td>USING USER02,R12</td>
<td>ROUTINE BASE</td>
<td>02250000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1880 USER02</td>
<td>DS</td>
<td>0H</td>
<td></td>
<td>02270000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001B8</td>
<td>D501 9704 C034 00704 001EC</td>
<td>1881</td>
<td>CLC</td>
<td>GDATE,X'7FFF' 7FFF?</td>
<td>02280000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001BE</td>
<td>4770 C026</td>
<td>001DE</td>
<td>1882</td>
<td>BNE</td>
<td>U29999 NO</td>
<td>02290000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001C2</td>
<td>955B 9728</td>
<td>00728</td>
<td>1883</td>
<td>CLI</td>
<td>GCHAR,C'='$' 1ST BYTE OF GCHAR=$? 02300000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001C6</td>
<td>4770 C026</td>
<td>001DE</td>
<td>1884</td>
<td>BNE</td>
<td>U29999 NO</td>
<td>02310000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001CA</td>
<td>D50F 9729 9728 00729 00728</td>
<td>1885</td>
<td>CLC</td>
<td>GCHAR+1(L'GCHAR-1),GCHAR REST=$?</td>
<td>02320000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001D0</td>
<td>4770 C026</td>
<td>001DE</td>
<td>1886</td>
<td>BNE</td>
<td>U29999 NO</td>
<td>02330000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001D4</td>
<td>D203 8009 C030 00009 001E8</td>
<td>1887</td>
<td>MVC</td>
<td>EXITEROR,=F'123456789' SET ERROR-CODE</td>
<td>02340000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001DA</td>
<td>9202 8008</td>
<td>00008</td>
<td>1888</td>
<td>MVI</td>
<td>EXITACTN,EXAWIERR FORCE ERROR</td>
<td>02350000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001DE</td>
<td>1890 U29999</td>
<td>DS</td>
<td>0H</td>
<td>02370000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001DE</td>
<td>BFEF 8003</td>
<td>00003</td>
<td>1891</td>
<td>ICM</td>
<td>R14,B'1111',EXITRETN GET RTN ADDRESS</td>
<td>02380000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LXTABTST - UMS SAMPLE LXTBL DEFINITION

LOC OBJECT CODE ADDR1 ADDR2 STMT STATEMENT
0001E2 07FE 1892 BR R14 EXIT 02390000
0001E8 1894 LTORG 02410000
0001E8 075BCD15 1895 =F'123456789'
0001EC 7FFF 1896 =X'7FFF'
1897 DROP 02420000
1899 END 02440000

Chapter Six - LXTABLE Processing 134
## Cross Reference Table

<table>
<thead>
<tr>
<th>Control Section</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>ORIGIN</td>
</tr>
<tr>
<td>LXTABTST</td>
<td>00</td>
</tr>
<tr>
<td>CSECT2</td>
<td>118</td>
</tr>
<tr>
<td>CSECT3</td>
<td>188</td>
</tr>
<tr>
<td>UGLTEST1</td>
<td>198</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Refers To Symbol</th>
<th>In Control Section</th>
<th>Location</th>
<th>Refers To Symbol</th>
<th>In Control Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>2F</td>
<td>CSECT2</td>
<td>CSECT2</td>
<td>47</td>
<td>CSECT2</td>
<td>CSECT2</td>
</tr>
<tr>
<td>70</td>
<td>CSECT2</td>
<td>CSECT2</td>
<td>74</td>
<td>CSECT2</td>
<td>CSECT2</td>
</tr>
<tr>
<td>B1</td>
<td>CSECT2</td>
<td>CSECT2</td>
<td>D3</td>
<td>CSECT2</td>
<td>CSECT2</td>
</tr>
<tr>
<td>EB</td>
<td>CSECT2</td>
<td>CSECT2</td>
<td>F9</td>
<td>CSECT2</td>
<td>CSECT2</td>
</tr>
<tr>
<td>111</td>
<td>CSECT2</td>
<td>CSECT2</td>
<td>1D</td>
<td>CSECT3</td>
<td>CSECT3</td>
</tr>
<tr>
<td>55</td>
<td>UGLTEST1</td>
<td>UGLTEST1</td>
<td>78</td>
<td>UGLTEST1</td>
<td>UGLTEST1</td>
</tr>
<tr>
<td>11C</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td>140</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
</tr>
<tr>
<td>148</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td>14C</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
</tr>
<tr>
<td>154</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td>158</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
</tr>
<tr>
<td>160</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td>164</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
</tr>
<tr>
<td>168</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td>170</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
</tr>
<tr>
<td>174</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td>178</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
</tr>
<tr>
<td>180</td>
<td>LXTABTST</td>
<td>LXTABTST</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ENTRY ADDRESS    00

TOTAL LENGTH    1F0
** LXTABTST REPLACED AND HAS AMODE 24
** LOAD MODULE HAS RMODE 24
** AUTHORIZATION CODE IS      0.
### Hexadecimal Dump of LXTABLE Example

AMASPZAP INSPECTS, MODIFIES, AND DUMPS CSECTS OR SPECIFIC DATA RECORDS ON DIRECT ACCESS STORAGE.

**DUMP LXTABTST ALL**

**CCHHR- 0011000314  RECORD LENGTH- 0001F0**

<table>
<thead>
<tr>
<th>MEMBER NAME</th>
<th>LXTABTST</th>
<th>CSECT NAME</th>
<th>LXTABTST</th>
</tr>
</thead>
<tbody>
<tr>
<td>F761F0F9</td>
<td>61F9F140</td>
<td>F1F54BF4</td>
<td>57000001</td>
</tr>
</tbody>
</table>

**07/09/91 15.47...**

<table>
<thead>
<tr>
<th>MEMBER NAME</th>
<th>LXTABTST</th>
<th>CSECT NAME</th>
<th>LXTABTST</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001180A</td>
<td>08080000</td>
<td>3C007109</td>
<td>090E0011</td>
</tr>
</tbody>
</table>

**CCHHR- 0011000314  RECORD LENGTH- 0001F0**

<table>
<thead>
<tr>
<th>MEMBER NAME</th>
<th>LXTABTST</th>
<th>CSECT NAME</th>
<th>CSECT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>7F61F0F9</td>
<td>61F9F140</td>
<td>F1F54BF4</td>
<td>57000001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MEMBER NAME</th>
<th>LXTABTST</th>
<th>CSECT NAME</th>
<th>CSECT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001180A</td>
<td>08080000</td>
<td>3C007109</td>
<td>090E0011</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MEMBER NAME</th>
<th>LXTABTST</th>
<th>CSECT NAME</th>
<th>LXTABTST</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001180A</td>
<td>08080000</td>
<td>3C007109</td>
<td>090E0011</td>
</tr>
</tbody>
</table>
000060  000000EF 08000008 00000107

**CCHHR-  0011000314   RECORD LENGTH-  0001F0   MEMBER NAME  LXTABTST  CSECT NAME  CSECT3
000000  000007B0 077C0201 0046079C 01

**CCHHR-  0011000314   RECORD LENGTH-  0001F0   MEMBER NAME  LXTABTST  CSECT NAME  UGLTEST1
000000  D60396C0 96C04780 C016910A 80074780  C0169201 8008BFEF 800307FE 00000000 *O...............*
000020  D5019704 C0344770 C026955B 97284770  C026D50F 97299728 4770C026 D2038009 *N..............$....*
000040  C0309202 8008BFEF 800307FE 00000000  075BCD15 7FFF

AMA113I COMPLETED DUMP REQUIREMENTS
AMA100I AMASPZAP PROCESSING COMPLETED

****************************************************************************** BOTTOM OF DATA ******************************************************************************
Resident Utilities

The Guest System Utilities are divided between non-resident and resident. The non-resident routines require access to large tables, such as the one that converts Massachusetts place names to a number and vice versa. Their functionality will be discussed later. The resident utilities are sections in the module UGZ0003P which is loaded into memory at system start-up. The addresses of the entry points for each of the routines in the module is placed in the Guest Common Area at transaction start-up by the UMS System Control Program, UGZ0001P.

This is another example of “soft linkage” between system and application. The utilities may be changed in virtually any way and the application will need no maintenance for it. In addition, where the utilities are not hard-linked to the application, the size of the application module is kept to a minimum. The linkage is achieved through a small routine, UICALLST, which must be hard-linked to the application program. Do not try any kind of direct calls. The coding is simple and is as follows for COBOL:

CALL 'UICALLST' USING <funcaddrfld> <fldlist>.

<funcaddrfld> parameter is the name of the Guest Common Area field designated for the service required. These begin at UGCOMMON-ASSIST-ADDRESS-AREA for COBOL and UGCAMOD1 in the Assembler version. The comments or the actual field name will designate the service provided.

 fldlist is the name(s) of the field(s) pertaining to the service called, see the detailed descriptions below.

Before issuing the call, one must properly initialize the Guest Common fields used by the particular routine. Each will be mentioned, in turn.
Date Conversion Routine

The caller provides one of the three date formats, Gregorian, Julian, or internal (two-byte binary) and the routine provides the other two. The choice is whether the user wants the day of the week also. The Gregorian date is returned as mm/dd/yyyy. It may be given as the 2 or 4 digit year, with or without the slashes, and the year may come first (yymmdd).

Note: The internal, binary or serial date is a two-byte binary value of the number of days since January 1, 1940.

The data areas of interest are:

1. The FORMAT byte: this designates the form of date given and whether or not to return day-of-week. Values allowed are 0-2 and 4-6.

<table>
<thead>
<tr>
<th>Return Day-of-week</th>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>input Binary</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>form Gregorian</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>given Julian</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

2. The return code or status byte has the following value:

- 0 = OK
- 1 = Range error
- 2 = Non-numeric input
- 3 = Invalid month
- 4 = Invalid day of month
- 5 = Invalid year
- 6 = Invalid day of year
- 7 = Null-date input
- 8 = Format unknown

Example of a COBOL Invocation.

CALL 'UICALLST' USING UGCOMMON-DATE-Routine ADDRESS DFHCOMMAREA.
The Date Data-Fields

COBOL

15 UGCOMMON-DATE-AREA.
20 UGCOMMON-DATE-INPUT-FORMAT    PIC X.
20 UGCOMMON-DATE-RETURN-STATUS    PIC X.
20 UGCOMMON-DATE-BINARY-FORMAT    PIC XX.
20 UGCOMMON-DATE-JULIAN-FORMAT    PIC 9(07).
20 FILLER REDEFINES UGCOMMON-DATE-JULIAN-FORMAT.
25 UGCOMMON-DATE-JULIAN-YEAR      PIC 9(04).
25 UGCOMMON-DATE-JULIAN-DAY       PIC 9(03).
20 UGCOMMON-DATE-GREG-FORMAT-X.
25 UGCOMMON-DATE-GREG-FORMAT      PIC 9(08).
20 FILLER REDEFINES UGCOMMON-DATE-GREG-FORMAT-X.
25 UGCOMMON-DATE-GREG-YEAR        PIC 9(04).
25 UGCOMMON-DATE-GREG-MONTH       PIC 9(02).
25 UGCOMMON-DATE-GREG-DAY         PIC 9(02).
20 UGCOMMON-DATE-DAY-OF-WEEK      PIC 9.

Assembler

UGCDCONA DS  0CL20     DATE CONVERSION AREA
UGCDCFMT DS  CL1       INPUT FORMAT
UGCDCSTA DS  CL1       RETURN STATUS
UGCDCBIN DS  XL2       BINARY FORMAT
UGCDCJUL DS  CL7       JULIAN YYYYDDD FORMAT
UGCDCGRE DS  CL8       GREGORIAN YYYYMMDD FORMAT
UGCDCDAY DS  CL1       DAY OF WEEK#

Data-Name Address Routine

This routine will provide full-word address (24-bit) address for a data field-name. The calling parameters are pairs of a field-name and (then) the name of the 4-byte field. The 4-byte-field is defined as S9(9) COMP, and receives the address. A sample COBOL invocation is:

05 FIELD-NAME-1       PIC X(20).
05 FIELD-NAME-2       PIC X.
10 FIELD-NAME-1-ADDR  PIC S9(9) COMP.
10 FIELD-NAME-2-ADDR  PIC S9(9) COMP.
CALL 'UICALLST' USING UGCOMMON-DATA-NANE-ADDRESS
    FIELD-NAME-1  FIELD-NAME-1-ADDR
    FIELD-NAME-2  FIELD-NAME-2-ADDR.

Upon return from the routine, the -ADDR locations hold the 24-bit address of their ‘companion fields.’ No return code is provided. If the field name is not entered first, before the full-word name in the call, the results are guaranteed to be unpredictable.

The third and fourth function fields in the list are currently unused and will, if used, give an immediate no-op return to your program. The fifth and sixth functions are tables for UMS system control program use only. If these functions are invoked, a program check will occur.

**Miscellaneous Edit Services**

The seventh function, UGCOMMON-MISC-TRANSLATE-TABLES, or UGCAMSTR in assembler provides some miscellaneous edits with Upper Case Translation. The COBOL calling sequence is:

```
CALL 'UICALLST' USING <comareaname><fldname>
```

- `<comareaname>` is the name of the start of the guest common area.
- `<fldname>` is the name of the field to be processed

Prior to issuing the call, one must prime UGEDCODE and UGEDLENG with the specific service code for the field and the field's length. In all of these services, the field’s characters are first translated from Lower to Upper case. The service codes, the implied field-lengths and the nature of the service are as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Length implied</th>
<th>Service description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
<td>Lower to Upper Case translation only</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>Mass. Lic.# edit (len optional)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>State-code edit (len optional)</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>Syntatic Lic#/State code edit (len optional)</td>
</tr>
</tbody>
</table>

Note: If the twos-compliment of the code is passed, the translate is skipped.
**In-Core Online Sort**

The eighth and last routine is the core sort. The limitation is that the table to be sorted size plus the **required** ‘dummy entry’ must not exceed 65,280 bytes. The routine is called with three parameters illustrated below for COBOL:

```
CALL 'UICALLST' USING <ugcaddr>, <sortarea>, <sortctrl>
```

- `<ugcaddr>` is the name of the start of the Common Area
- `<sortarea>` is the name of the sort area or table to be sorted as illustrated below
- `<sortctrl>` is the name of a 12 byte sort-descriptor data structure illustrated below

```
01 SORT-TABLE.
  05 ST-ELEMENT OCCURS XX TIMES PIC X(YY).
  05 ST-DUMMY PIC X(YY).

01 SORT-DESCRIPTOR.
  05 SD-NUM-ENTRIES PIC S9(9) COMP.
      (not including the 'dummy entry')
  05 SD-LEN-ENTRY PIC S9(4) COMP.
  88 SD-GOOD-LENGTH VALUE 1 THRU 256.
  05 SD-LEN-KEY PIC S9(4) COMP.
  05 SD-POSITION-OF-KEY PIC S9(4) COMP.
  05 SD-SORT-TYPE PIC X.
      88 SD-ASCENDING VALUE 'A'.
      88 SD-DESCENDING VALUE 'D'.
  05 SD-STATUS-BINARY PIC X.
      (NOTE: 88 SD-STATUS-OK VALUE 0.
       these 88 SD-STATUS-BAD-TYPE VALUE 4.
       are 88 SD-STATUS-BAD-ENTRY-LEN VALUE 8.
       to 88 SD-STATUS-BAD-KEY-LEN VALUE 12.
       document 88 SD-STATUS-BAD-KEY-POSN VALUE 16.
       values 88 SD-STATUS-TOO-MANY-ENTRIES VALUE 20.
       only) 88 SD-STATUS-TABLE-TOO-LARGE VALUE 24.

To inspect the status in COBOL try:

```
01 SORT-STATUS-WORK PIC S9(4) COMP.
01 FILLER REDEFINES SORT-STATUS-WORK.
  05 FILLER PIC X.
  05 SS-WORK PIC X.
```
Non-Resident Utilities

The non-resident utilities are a group of services whose satisfaction depends on entities that are not necessarily present in memory. Fulfillment of the service may require CICS Loads for tables or file reads to accumulate the needed resources. A separate program, UGZ0024P, manages these services on the Guest Side. The functions or services provided concern the Violation Table, the Court (Information) Table and the expansion of USH Lines. They are:

<table>
<thead>
<tr>
<th>Function Id</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00'</td>
<td>Return whole VTAB entry</td>
</tr>
<tr>
<td>X'01'</td>
<td>Return VTAB Long Description</td>
</tr>
<tr>
<td>X'02'</td>
<td>Return VTAB Short Description</td>
</tr>
<tr>
<td>X'03'</td>
<td>Return Court Information</td>
</tr>
<tr>
<td>X'04'</td>
<td>Expand Compressed USH Lines</td>
</tr>
</tbody>
</table>

Violation Table Processing

The parameter list requires the half-word request code or function-id, the half-word violation surrogate number and the address of the field to receive the data returned. If a request code of x'00' is passed, a 115 Byte structure is returned as shown below, otherwise only the requested description is returned.

```assembly
01 VIOLATION-TABLE-RECORD.
   05 VTR-SURROGATE        PIC S9(4) COMP.
   05 VTR-CODE             PIC X(9).
   05 VTR-FEDERAL-VIOLATION PIC XXX.
   05 VTR-LONG-DESCRIPTION PIC X(40).
   05 VTR-SHORT-DESCRIPTION PIC X(20).
   05 VTR-ACTION-GROUP     OCCURS 4 TIMES.
      10 VTR-ACTION         PIC X.
      10 VTR-ACTION-SUBJECT PIC X.
```
10 VTR-ACTION-DISCRETIONARY PIC X.
10 VTR-ACTION-PHYS PIC S9(4) COMP.
10 VTR-QNTY-PERIOD PIC 9(5) COMP-3.
05 VTR-INDC-CRIMINAL PIC X.
  88 VTR-INDC-NON-CRIMINAL VALUE 'N'.
  88 VTR-INDC-CRIMINAL VALUE 'C'.
05 VTR-INDC-MAJOR-MINOR PIC X.
  88 VTR-INDC-MAJOR-VIOLATION-88 VALUE '1'.
  88 VTR-INDC-MINOR-VIOLATION-88 VALUE '0'.
05 VTR-QNTY-GRACE PIC 99.
05 VTR-QNTY-HTO-WEIGHT PIC 9 COMP-3.
05 VTR-QNTY-POINTS-MAX PIC 9 COMP-3.
05 VTR-QNTY-TIME-SPAN PIC S9(4) COMP.
05 VTR-CODE-VIOL-TYPE PIC X.
  88 VTR-CODE-OWNER-88 VALUE '0'.
  88 VTR-CODE-OPER-88 VALUE '1'.
  88 VTR-CODE-OWN-OP-88 VALUE '2'.

Code Fragment Example (from UGL0011P)

05 WS-VTAB-AREA.
  10 WS-VTAB-FUNCTION PIC S9(4) COMP.
  10 WS-VTAB-SURR PIC S9(4) COMP.
  10 WS-VTAB-RETURN-ADDR PIC S9(9) COMP.

MOVE +1 TO WS-VTAB-FUNCTION.
MOVE WORK-VTAB-NUMB-SURR TO WS-VTAB-SURR.
CALL 'UICALLST' USING UGCOMMON-DATA-NAME-ADDRESS
        UGCOMMON-ERROR-TEXT
        WS-VTAB-RETURN-ADDRESS.

The above call generates the address for the VTAB text
return area. This address is required for the next
Non-resident Service Call.

MOVE WS-VTAB-AREA TO UGCOMMON-DOUBLE-WORD-TEMP.
EXEC CICS LINK PROGRAM (UGCOMMON-NONRES-SERVICES-MOD)
        COMMAREA (DFHCOMMAREA)
        LENGTH (UGCOMMON-XCTL-LENGTH)
END-EXEC.
Court Table Processing

The request for Court Information will be similar to that for the Violation Table. The request code of X'03' returns a 71-byte structure as follows:

```plaintext
01 COURT-RECORD.
  05 COURT-CODE PIC 999.
  05 COURT-CITY PIC X(15).
  05 COURT-STATE PIC XX.
  05 COURT-STREET PIC X(20).
  05 COURT-ZIP PIC X(6).
  05 COURT-NAME PIC X(25).
```

Compressed USH Lines Processing

This example is drawn from UGS0101P. It is similar to the above example except that the parameter fields are in a redefine of the commarea so there is no group parameter move to the commarea before the call to non-resident services.

```plaintext
01 FILLER REDEFINES DFHCOMMAREA.
  05 NRS-FCODE PIC S9(4) COMP.
  05 NRS-PARM1 PIC S9(4) COMP.
  05 NRS-PARM2 PIC S9(9) COMP.
  05 FILLER PIC X(4066).

1400-CALL-FOR-DETAIL.
  CALL 'UICALLST' USING UGCOMMON-DATA-NAME-ADDRESS
    UGS0100M-LASTBYTE-PLUS-1
    1WORD-TEMP.
  COMPUTE UGCOMMON-CURRENT-MAP-LENGTH =
    1WORD-TEMP - MAP-POINTER.
  MOVE +4 TO NRS-FCODE.
  CALL 'UICALLST' USING UGCOMMON-DATA-NAME-ADDRESS
    UGS0100M-MDETL14-G
    NRS-PARM2.
  IF USH-HEADER-REQUIRED
    CALL 'UICALLST' USING UGCOMMONDATA-NAME-ADDRESS
      USH-DETAIL-DATA
      1WORD-TEMP
  ELSE
```
CALL 'UICALLST' USING UGCOMMONDATA-NAME-ADDRESS
   USH-DETAIL-ONLY-DATA
   1WORD-TEMP.

NOTE: USH-HEADER-REQUIRED is an 88-level on the byte immediately preceding USH-DETAIL-DATA. A 'Y' asks the header and a 'N' says 'no'. This value must be set before the call to NONRES services.

COMPUTE NRS-PARM1 =
   1WORD-TEMP - UGCOMMON-HOST-DATA-AREA-ADDR.
EXEC CICS LINK PROGRAM (UGCOMMON-NONRES-SERVICES-MOD)
   COMMAREA (DFHCOMMAREA)
   LENGTH (UGCOMMON-XCTL-LENGTH)

END-EXEC.
CICS Tables

Under CICS, a task is an execution of one or more programs which function together as a unit called a transaction. When a transaction is invoked, a specified application program is loaded into storage (if it is not already) and a task is started. A special CICS table, called the Processing Program Table (PPT), contains a list of all the valid program names, LX tables and maps that may be executed. LX tables are unique to UMS and each is a macro that performs field editing and mapping for a UMS screen. Each CICS region has its own PPT. The PPT indicates to CICS the location of each program, LX table and map, with a storage address if it has already been loaded, or a disk location if it has not been loaded. CICS uses the PPT to determine whether it will load a new copy of the program, LX table or map into virtual storage when the transaction is invoked.

Each time a new version of a program, LX table or map is created, it must be added to the PPTs in each CICS region. Following are the formats of the two DFHPPT macro entries in a PPT that are used to define a program, LX table or map:

```
  COLUMN COLUMN COLUMN
1   10  45   72
   |   |   |   |
name DFHPPT TYPE=ENTRY, description of entry X
      PROGRAM=name,       X
      PGMLANG=COBOL

name DFHPPT TYPE=ENTRY, description of entry X
      PROGRAM=name
```

The first format is used for COBOL programs, as indicated by the PGMLANG parameter. When this parameter is omitted, as in the second format, then the program language for that PPT entry is assumed to be assembler language. This second entry format is used for LX tables and maps. The field labeled “name,” on both formats, is where the programmer specifies the name of the program, LX table or map. This name must correspond to the member name of the module in the on-line CICS load library. To the right of either entry format is a brief description of the entry. The “X” in column 72 is a continuation character and signifies that the line continues onto the next line. The
following are examples of PPT entries for the program UGU1041P, the map UGU1040M
and the LX table UGU1040T, which are used for the UPH screen:

<table>
<thead>
<tr>
<th>COLUMN</th>
<th>COLUMN</th>
<th>COLUMN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>45</td>
</tr>
</tbody>
</table>

| UGU1041P | DFHPPT TYPE=ENTRY, | UMS UPH GUEST INPUT | X |
|          | PROGRAM=UGU1041P,  |                      | X |
|          | PGMLANG=COBOL      |                      |   |
| UGU1040M | DFHPPT TYPE=ENTRY, | UMS UPH INQ SCREEN  | X |
|          | PROGRAM=UGU1040M   |                      |   |
| UGU1040T | DFHPPT TYPE=ENTRY, | UMS UPH INQUIRY LXTABLE | X |
|          | PROGRAM=UGU1040T   |                      |   |

When an SE wishes to add, change or delete PPT entries, the SE will need to create a
TSO member using a mask. The library that contains all the necessary information is
called RMVOL.JMV.TBLLIB. In this library, the member “HELP” contains instructions
on filling out a mask. The mask can then be filled out by using the member
“CICSMASK,” also in RMVOL.JMV.TBLLIB, and then saving the updated mask into
RMVOL.JMV.TBLLIB under a new name. The new member name should follow the
naming convention “PPTXXXXX” where PPT specifies the PPT table and “XXXXX” is
a set of characters the SE supplies to make the member name unique. To create a mask,
the SE fills out the requestor information fields in “CICSMASK,” appends his PPT
entries to the end of the member, and saves it under the new name. The requestor
information fields include name, phone number, an indication of whether this is an add,
delete or change, a list of the CICS regions to update, an indicator for necessary JCL
changes, and comments.

The mask member may be used to update entries in other CICS tables, such as the
Program Control Table (PCT) and the File Control Table (FCT). PPT entries are the only
entries discussed in this documentation section because they are the only ones that apply
for versioning.

When the updated mask member has been completed and saved into
RMVOL.JMV.TBLLIB, the SE should call the CICS Support Group in Boston to inform
them of the member name. CICS support will review the member and discuss what
needs to be done at that time. This will ensure that they understand what updates to
perform to which CICS regions and when the changes need to take place.
For PPT entries that need to be updated in CV10, the programmer should make arrangements for a CICS administrator to enter the updates on-line. The update mask must be filled out for all the other regions.

Changes are applied to the test and production systems on a weekly basis. The cutoff date/time for the test regions (T1, S1 and R1) is 3 P.M. on Monday. The updates will become effective following test CICS maintenance at noon on Tuesday. The cutoff date/time for the production regions (P1, A1 and M1) for MMS move only is Tuesday. The updates will become effective following production CICS maintenance on Wednesday.

After the updates have taken place, the SE should verify that the PPT updates were successful. To verify successful completion of updates (adds) for UGU1040M and UGU1040T, the SE would key in the following and then press enter:

CEMT IN PR(UGU1040*)

The “*” in the example is a wild card character that means any character is valid in that position. If the add was successful, the following would be displayed:

PRO(UGU1040M) LEN(0000000) RECS(000) USE(000000) ASS ENA
PRO(UGU1040T) LEN(0000000) RECS(000) USE(000000) ASS ENA

The character string “ASS” signifies that the program language is assembler language (“COB” would signify COBOL), and “ENA” says that it is enabled. The SE must verify all updates. F3 terminates the “CEMT” transaction. In some regions, the SE may not have security to execute the “CEMT” transaction. In these cases, he/she should call the CICS Support Group in Boston and have them perform a PPT table lookup to verify the updates.

When a new version of a UMS program, LX table, or map is placed in the PPTs, the SE should have the old version name deleted from the PPTs. This should only be done when the new version has been in production a sufficient amount of time and the old version is no longer being executed. This way, the PPT tables will reflect current programs, LX tables, and maps and will not fill up with obsolete entries.

**UMS Setup Options UGZ0014P**

The application portion of UMS requires a minimum of two TRANIDs to execute. These are referred to as a session-startup TRANID and a default session-run TRANID. The default values are UG03 and UG04 respectively. A site may elect to run with alternative TRANID’s. To do this, two actions are required. First, PCT entries must be setup for the desired TRANID’s. These should point to UGZ0001P, just as the release TRANID’s do.
Secondly, the desired TRANID's need be specified in the SETUP macro within UGZ0014P, and this module must be reassembled. The symbolic names are TRAN1 and TRAN2. Thus, to make the session-startup TRANID be ZZZZ and the session-run TRANID be AAAA, the following would be required:

.....,TRAN1=ZZZZ,TRAN2=AAAA...

Note that the value of TRAN1 must be 4 EBCDIC characters, but the value of TRAN2 may be 4 EBCDIC characters or 8 hexadecimal characters. This allows making the default session-run TRANID a value which cannot be entered by a terminal operator very easy. You should also note that UMS business functions normally all execute under the default session-run TRANID. If, for some site dependent reason this is not satisfactory, the site may specify alternative TRANID's for some (or all) functions. This is accomplished by generating the requisite PCT entries (identical to the default UG04 entry), and reassembling UGZ0002P after including the desired TRANID on the optional TASK= parameter of each PCTEG you desire to change. The allowable values are 4 EBCDIC characters or 8 hexadecimal characters.

UMS contains a “hotkey” option, which allows the user to enter a second logical session, for inquiry purposes (presumably related to some update function being attempted in the primary session). The term “hotkey” refers to the key used to toggle between the sessions. The default key used is PA1. The site may elect to use any of the three PA keys for this purpose. This is specified in the SETUP macro with the HOTKEY symbolic. The allowable values are PA1, PA2, or PA3. If you desired to use PA2, the entry would be:

.............,HOTKEY=PA2

The use of the HOTKEY option by a user requires the UMS software to save the current user environment so that it can be restored when the user returns to the primary logical session. The environment is saved in CICS temporary storage, as two records on a queue named UMSQ????, where ???? is the involved TERMID. The total length of the two records is a minimum of 4096 bytes, and a maximum of approximately 6500 bytes, depending largely on the amount of data currently being displayed in the primary session. The queue is deleted on return from the secondary to the primary session. By default, UMS will use TEMP STORAGE AUX. If the site desires, performance might be enhanced by the use of TEMP STORAGE MAIN. This is specified in the SETUP macro by specifying a value of MAIN for the symbolic TEMPSTOR:

.............,TEMPSTOR=MAIN

This option should ONLY be used if your CICS version is at least 1.7 and your MVS version is at least XA.
As discussed previously, UMS provides two internal mechanisms for preserving its COMMAREA across pseudo-conversational interactions. One of these ways operates outside of traditional CICS technique, the other uses traditional CICS techniques and a VSAM cluster. Both of these techniques account for limitations inherent in older CICS and MVS versions. UMS also supports a mechanism which allows a more conventional CICS preservation of COMMAREA’s. This mechanism is only available if your CICS version is at least 1.7 and your MVS version is at least XA. To use this option, you must specify the value of the SYSTEM symbolic as XA and the value of the COMSAVE symbolic as CICS:

......SYSTEM=XA,COMSAVE=CICS

You must still specify a device list in UGZ0014P, but the list need only contain 1 device and the device does not have to exist. Note that by electing this option, you lose the feature of being able to restrict the UMS access via the UGZ0014P device list. Also, as compared with the UMS high-core save option, this technique has a higher degree of system overhead.

Comparison of UMS COMMAREA save options:

<table>
<thead>
<tr>
<th></th>
<th>MVS-SP</th>
<th>MVS-XA/ESA</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS 1.6.1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>CICS 1.7 or</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>better</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. must use SYSTEM=SP (VSAM swap)

2. may use SYSTEM=XA (highcore swap) or SYSTEM=SP (VSAM swap)  
   SYSTEM=XA is recommended

3. may use SYSTEM=XA (highcore swap), SYSTEM=SP (VSAM swap) or  
   SYSTEM=XA,COMSAVE=CICS

   SYSTEM=SP is not recommended

   SYSTEM=XA,COMSAVE=CICS is recommended if you use autoinstall  
   with random id generation, or have a very large  
   terminal base with sparse use of UMS

   SYSTEM=XA is recommended in all other cases
GROUP NAME: PCTUMSGM

**************

TRANSACTIONS:
UGCL    UGCM    UGCY    UGLD    UGPR    UGSA
UGXX    UGYY    UG01    UG03    UG04

CICS/VS  RDO OFF-LINE UTILITY PROGRAM

*****************************************************************

OBJECTS IN GROUPS UTILITY CO

*****************************************************************

GROUP NAME: PCTUMSGM

**************

TRANSACTION (UGCL)    GROUP (PCTUMSGM) PROGRAM (UGZCLERP)    TWASIZE(0)
TRANSACTION (UGCM)    GROUP (PCTUMSGM) PROGRAM (UCOREMTR)    TWASIZE(0)
TRANSACTION (UGCY)    GROUP (PCTUMSGM) PROGRAM (UGZ0027P)    TWASIZE(0)
TRANSACTION (UGLD)    GROUP (PCTUMSGM) PROGRAM (UGZLOADP)    TWASIZE(0)
TRANSACTION (UGPR)    GROUP (PCTUMSGM) PROGRAM (UGZ0030P)    TWASIZE(0)
TRANSACTION (UGSA)    GROUP (PCTUMSGM) PROGRAM (USAMPLEP)    TWASIZE(0)
TRANSACTION (UGTH)    GROUP (PCTUMSGM) PROGRAM (UGZ0020P)    TWASIZE(0)
TRANSACTION (UGTL)    GROUP (PCTUMSGM) PROGRAM (UGZ0015P)    TWASIZE(0)
TRANSACTION (UGTM)  GROUP (PCTUMSGM)  PROGRAM (UGZ0990P)  TWASIZE (0)
TRANSACTION (UGXX)  GROUP (PCTUMSGM)  PROGRAM (UGZXXXXP)  TWASIZE (0)
TRANSACTION (UGYY)  GROUP (PCTUMSGM)  PROGRAM (UGZYYYYP)  TWASIZE (0)
TRANSACTION (UG01)  GROUP (PCTUMSGM)  PROGRAM (UGT0001P)  TWASIZE (0)
TRANSACTION (UG03)  GROUP (PCTUMSGM)  PROGRAM (UGZ0001P)  TWASIZE (0)
TRANSACTION (UG04)  GROUP (PCTUMSGM)  PROGRAM (UGZ0001P)  TWASIZE (0)

GROUP NAME: PPTUMSGM

UGB0010M  UGB0340P  UGB3000P  UGC1062P  UGL0220M  UGL0330P  UGL0580P  UGL1201P
UGB0010P  UGB0340T  UGB3000T  UGETVOLS  UGL0220P  UGL0330T  UGL0580T  UGL3200M
UGB0010T  UGB0341P  UGB3001P  UGI1320M  UGL0220T  UGL0331P  UGL0600M  UGL3200P
UGB0011P  UGB0410M  UGB3020M  UGI1320T  UGL0221P  UGL0340M  UGL0600P  UGL3200T
UGB0020M  UGB0410P  UGB3020P  UGI1321P  UGL0240M  UGL0340P  UGL0600T  UGL3201P
UGB0020P  UGB0410T  UGB3020T  UGI1322P  UGL0240P  UGL0340T  UGL0601P  UGL3206M
UGB0020T  UGB0411P  UGB3021P  UGI1340M  UGL0240T  UGL0341P  UGL0620M  UGL3206P
UGB0021P  UGB0420M  UGB3040M  UGI1340T  UGL0241P  UGL0350M  UGL0620P  UGL3206T
UGB0022P  UGB0420P  UGB3040P  UGI1341P  UGL0260M  UGL0350P  UGL0620T  UGL3207P
UGB0030M  UGB0420T  UGB3040T  UGI1342P  UGL0260P  UGL0350T  UGL0621P  UGL3300M
UGB0030P  UGB0421P  UGB3041P  UGI1350M  UGL0260T  UGL0351P  UGL0640M  UGL3300P
UGB0030T  UGB0430M  UGB3060M  UGI1350T  UGL0261P  UGL0400M  UGL0640P  UGL3300T
UGB0031S  UGB0430P  UGB3060P  UGI1351P  UGL0262P  UGL0400P  UGL0640T  UGL3301P
UGB0031P  UGB0430T  UGB3060T  UGI1352P  UGL0270M  UGL0400T  UGL0641P  UGL3320M
UGB0040M  UGB0431P  UGB3061P  UGL0010M  UGL0270P  UGL0401P  UGL0650M  UGL3320P
UGB0040P  UGB0500M  UGC0010M  UGL0010P  UGL0270T  UGL0410M  UGL0650P  UGL3320T
<table>
<thead>
<tr>
<th>UGB0040T</th>
<th>UGB0500P</th>
<th>UGC0010T</th>
<th>UGL0011P</th>
<th>UGL0271P</th>
<th>UGL0410P</th>
<th>UGL0650T</th>
<th>UGL3321P</th>
</tr>
</thead>
<tbody>
<tr>
<td>UGB0041P</td>
<td>UGB0500T</td>
<td>UGC0011P</td>
<td>UGL0020M</td>
<td>UGL0280M</td>
<td>UGL0410T</td>
<td>UGL0651P</td>
<td>UGL3430P</td>
</tr>
<tr>
<td>UGB0100P</td>
<td>UGB0501P</td>
<td>UGC0012P</td>
<td>UGL0020P</td>
<td>UGL0280P</td>
<td>UGL0411P</td>
<td>UGL0700M</td>
<td>UGL3600P</td>
</tr>
<tr>
<td>UGB0210M</td>
<td>UGB0700M</td>
<td>UGC0610M</td>
<td>UGL0021P</td>
<td>UGL0280T</td>
<td>UGL0480M</td>
<td>UGL0700P</td>
<td>UGM0001M</td>
</tr>
<tr>
<td>UGB0210P</td>
<td>UGB0700P</td>
<td>UGC0610T</td>
<td>UGL0030M</td>
<td>UGL0281P</td>
<td>UGL0480P</td>
<td>UGL0700T</td>
<td>UGM0001P</td>
</tr>
<tr>
<td>UGB0210T</td>
<td>UGB0700T</td>
<td>UGC0611P</td>
<td>UGL0030P</td>
<td>UGL0290M</td>
<td>UGL0480T</td>
<td>UGL0701P</td>
<td>UGM0001T</td>
</tr>
<tr>
<td>UGB0211P</td>
<td>UGB0701P</td>
<td>UGC0612P</td>
<td>UGL0031P</td>
<td>UGL0290P</td>
<td>UGL0481P</td>
<td>UGL0870M</td>
<td>UGM0002P</td>
</tr>
<tr>
<td>UGB0212P</td>
<td>UGB0702K</td>
<td>UGC1030M</td>
<td>UGL0040M</td>
<td>UGL0290T</td>
<td>UGL0500M</td>
<td>UGL0870P</td>
<td>UGM0010M</td>
</tr>
<tr>
<td>UGB0213P</td>
<td>UGB0702P</td>
<td>UGC1030T</td>
<td>UGL0040P</td>
<td>UGL0291P</td>
<td>UGL0500P</td>
<td>UGL0870T</td>
<td>UGM0010P</td>
</tr>
<tr>
<td>UGB0310M</td>
<td>UGB0710M</td>
<td>UGC1031P</td>
<td>UGL0060M</td>
<td>UGL0292M</td>
<td>UGL0500T</td>
<td>UGL0871P</td>
<td>UGM0010T</td>
</tr>
<tr>
<td>UGB0310P</td>
<td>UGB0710P</td>
<td>UGC1032P</td>
<td>UGL0060P</td>
<td>UGL0292P</td>
<td>UGL0510M</td>
<td>UGL1000P</td>
<td>UGM0011T</td>
</tr>
<tr>
<td>UGB0310T</td>
<td>UGB0710T</td>
<td>UGC1040M</td>
<td>UGL0061P</td>
<td>UGL0292T</td>
<td>UGL0510P</td>
<td>UGL1001P</td>
<td>UGM0012M</td>
</tr>
<tr>
<td>UGB0311P</td>
<td>UGB0711P</td>
<td>UGC1040T</td>
<td>UGL0062M</td>
<td>UGL0293P</td>
<td>UGL0510T</td>
<td>UGL1007P</td>
<td>UGM0012P</td>
</tr>
<tr>
<td>UGB0320M</td>
<td>UGB0720M</td>
<td>UGC1041P</td>
<td>UGL0062P</td>
<td>UGL0310M</td>
<td>UGL0540M</td>
<td>UGL1008P</td>
<td>UGM0012T</td>
</tr>
<tr>
<td>UGB0320P</td>
<td>UGB0720P</td>
<td>UGC1042P</td>
<td>UGL0070M</td>
<td>UGL0310P</td>
<td>UGL0540P</td>
<td>UGL1009P</td>
<td>UGM0013P</td>
</tr>
<tr>
<td>UGB0320T</td>
<td>UGB0720T</td>
<td>UGC1050M</td>
<td>UGL0070P</td>
<td>UGL0310T</td>
<td>UGL0540T</td>
<td>UGL1070M</td>
<td>UGM0020M</td>
</tr>
<tr>
<td>UGB0321P</td>
<td>UGB0721P</td>
<td>UGC1050T</td>
<td>UGL0071P</td>
<td>UGL0311P</td>
<td>UGL0541P</td>
<td>UGL1070P</td>
<td>UGM0020P</td>
</tr>
<tr>
<td>UGB0330M</td>
<td>UGB0730M</td>
<td>UGC1051P</td>
<td>UGL0200M</td>
<td>UGL0320M</td>
<td>UGL0560M</td>
<td>UGL1070T</td>
<td>UGM0020T</td>
</tr>
<tr>
<td>UGB0330P</td>
<td>UGB0730P</td>
<td>UGC1052P</td>
<td>UGL0200P</td>
<td>UGL0320P</td>
<td>UGL0560P</td>
<td>UGL1071P</td>
<td>UGM0021M</td>
</tr>
<tr>
<td>UGB0330T</td>
<td>UGB0730T</td>
<td>UGC1060M</td>
<td>UGL0200T</td>
<td>UGL0320T</td>
<td>UGL0560T</td>
<td>UGL1200M</td>
<td>UGM0021P</td>
</tr>
<tr>
<td>UGB0331P</td>
<td>UGB0731P</td>
<td>UGC1060T</td>
<td>UGL0201P</td>
<td>UGL0321P</td>
<td>UGL0561P</td>
<td>UGL1200P</td>
<td>UGM0021T</td>
</tr>
<tr>
<td>UGB0340M</td>
<td>UGB3000M</td>
<td>UGC1061P</td>
<td>UGL0202P</td>
<td>UGL0330M</td>
<td>UGL0580M</td>
<td>UGL1200T</td>
<td>UGM0022M</td>
</tr>
<tr>
<td>UGM0022P</td>
<td>UGM0062P</td>
<td>UGN0060M</td>
<td>UGN0280P</td>
<td>UGR0110P</td>
<td>UGR1092P</td>
<td>UGR1250M</td>
<td>UGR2201P</td>
</tr>
<tr>
<td>UGM0022T</td>
<td>UGM0063P</td>
<td>UGN0060P</td>
<td>UGN0280T</td>
<td>UGRX100T</td>
<td>UGR1100M</td>
<td>UGR1250T</td>
<td>UGR2202P</td>
</tr>
<tr>
<td>UGM0023P</td>
<td>UGM0064M</td>
<td>UGN0060T</td>
<td>UGN0300M</td>
<td>UGRX101T</td>
<td>UGR1140M</td>
<td>UGR1251P</td>
<td>UGR2260M</td>
</tr>
<tr>
<td>UGM0024P</td>
<td>UGM0064P</td>
<td>UGN0061P</td>
<td>UGN0300P</td>
<td>UGRX102T</td>
<td>UGR1140T</td>
<td>UGR1252P</td>
<td>UGR2260T</td>
</tr>
<tr>
<td>UGM0025P</td>
<td>UGM0066P</td>
<td>UGN0080M</td>
<td>UGN0300T</td>
<td>UGRX110T</td>
<td>UGR1141P</td>
<td>UGR1260M</td>
<td>UGR2261P</td>
</tr>
<tr>
<td>UGM0026P</td>
<td>UGM0066T</td>
<td>UGN0080P</td>
<td>UGN0301P</td>
<td>UGR0010M</td>
<td>UGR1142P</td>
<td>UGR1260T</td>
<td>UGR2262P</td>
</tr>
<tr>
<td>UGM0027P</td>
<td>UGM0067P</td>
<td>UGN0080T</td>
<td>UGN0320M</td>
<td>UGR0010P</td>
<td>UGR1170M</td>
<td>UGR1261P</td>
<td>UGR2300M</td>
</tr>
<tr>
<td>UGM0028P</td>
<td>UGM0068P</td>
<td>UGN0081P</td>
<td>UGN0320P</td>
<td>UGR0011P</td>
<td>UGR1170T</td>
<td>UGR1262P</td>
<td>UGR2300T</td>
</tr>
<tr>
<td>UGM0029P</td>
<td>UGM0068T</td>
<td>UGN0100M</td>
<td>UGN0320T</td>
<td>UGR0020M</td>
<td>UGR1171P</td>
<td>UGR1270M</td>
<td>UGR2301P</td>
</tr>
<tr>
<td>UGM0030M</td>
<td>UGM0069P</td>
<td>UGN0120M</td>
<td>UGN0321P</td>
<td>UGR0020P</td>
<td>UGR1172P</td>
<td>UGR1270T</td>
<td>UGR2302P</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>UGM0030P</td>
<td>UGM0070M</td>
<td>UGN0120P</td>
<td>UGN0322P</td>
<td>UGR0020T</td>
<td>UGR1180M</td>
<td>UGR1271P</td>
<td>UGR3060M</td>
</tr>
<tr>
<td>UGM0030T</td>
<td>UGM0070P</td>
<td>UGN0120T</td>
<td>UGN0360M</td>
<td>UGR0021P</td>
<td>UGR1180T</td>
<td>UGR1272P</td>
<td>UGR3060T</td>
</tr>
<tr>
<td>UGM0031P</td>
<td>UGM0070T</td>
<td>UGN0121P</td>
<td>UGN0360P</td>
<td>UGR0310M</td>
<td>UGR1181P</td>
<td>UGR1280M</td>
<td>UGR3061P</td>
</tr>
<tr>
<td>UGM0032M</td>
<td>UGM0071P</td>
<td>UGN0140M</td>
<td>UGN0360T</td>
<td>UGR0310T</td>
<td>UGR1182P</td>
<td>UGR1280T</td>
<td>UGR3062P</td>
</tr>
<tr>
<td>UGM0032P</td>
<td>UGM0080M</td>
<td>UGN0140P</td>
<td>UGN0361P</td>
<td>UGR0311P</td>
<td>UGR1190M</td>
<td>UGR1281P</td>
<td>UGR3260M</td>
</tr>
<tr>
<td>UGM0032T</td>
<td>UGM0080P</td>
<td>UGN0140T</td>
<td>UGN0420M</td>
<td>UGR0312P</td>
<td>UGR1190T</td>
<td>UGR1282P</td>
<td>UGR3260T</td>
</tr>
<tr>
<td>UGM0033P</td>
<td>UGM0080T</td>
<td>UGN0141P</td>
<td>UGN0420P</td>
<td>UGR0540P</td>
<td>UGR1191P</td>
<td>UGR1310M</td>
<td>UGR3261P</td>
</tr>
<tr>
<td>UGM0040M</td>
<td>UGM0081P</td>
<td>UGN0160M</td>
<td>UGN0420T</td>
<td>UGR0541P</td>
<td>UGR1192P</td>
<td>UGR1310T</td>
<td>UGR3262P</td>
</tr>
<tr>
<td>UGM0040P</td>
<td>UGM0090M</td>
<td>UGN0160P</td>
<td>UGN0421P</td>
<td>UGR0560P</td>
<td>UGR1200M</td>
<td>UGR1311P</td>
<td>UGR4060M</td>
</tr>
<tr>
<td>UGM0040T</td>
<td>UGM0090P</td>
<td>UGN0160T</td>
<td>UGN0430M</td>
<td>UGR0561P</td>
<td>UGR1200T</td>
<td>UGR1312P</td>
<td>UGR4060T</td>
</tr>
<tr>
<td>UGM0041P</td>
<td>UGM0090T</td>
<td>UGN0161P</td>
<td>UGN0430P</td>
<td>UGR1000M</td>
<td>UGR1201P</td>
<td>UGR2060M</td>
<td>UGR4061P</td>
</tr>
<tr>
<td>UGM0050M</td>
<td>UGM0091M</td>
<td>UGN0180M</td>
<td>UGN0430T</td>
<td>UGR1010M</td>
<td>UGR1202P</td>
<td>UGR2060T</td>
<td>UGR4062P</td>
</tr>
<tr>
<td>UGM0050P</td>
<td>UGM0091P</td>
<td>UGN0180P</td>
<td>UGN0431P</td>
<td>UGR1020M</td>
<td>UGR1210M</td>
<td>UGR2061P</td>
<td>UGR5060M</td>
</tr>
<tr>
<td>UGM0050T</td>
<td>UGM0092M</td>
<td>UGN0180T</td>
<td>UGN0440M</td>
<td>UGR1060M</td>
<td>UGR1210T</td>
<td>UGR2062P</td>
<td>UGR5060T</td>
</tr>
<tr>
<td>UGM0051P</td>
<td>UGM0092P</td>
<td>UGN0181P</td>
<td>UGN0440P</td>
<td>UGR1060T</td>
<td>UGR1211P</td>
<td>UGR2140M</td>
<td>UGR5061P</td>
</tr>
<tr>
<td>UGM0052M</td>
<td>UGM0093M</td>
<td>UGN0200M</td>
<td>UGN0440T</td>
<td>UGR1061P</td>
<td>UGR1212P</td>
<td>UGR2140T</td>
<td>UGR5062P</td>
</tr>
<tr>
<td>UGM0052P</td>
<td>UGM0094M</td>
<td>UGN0200P</td>
<td>UGN0441P</td>
<td>UGR1062P</td>
<td>UGR1220M</td>
<td>UGR2141P</td>
<td>UGS0100M</td>
</tr>
<tr>
<td>UGM0052T</td>
<td>UGM0094P</td>
<td>UGN0200T</td>
<td>UGN0460M</td>
<td>UGR1070M</td>
<td>UGR1220T</td>
<td>UGR2142P</td>
<td>UGS0100P</td>
</tr>
<tr>
<td>UGM0053P</td>
<td>UGM0094T</td>
<td>UGN0201P</td>
<td>UGN0460P</td>
<td>UGR1070P</td>
<td>UGR1221P</td>
<td>UGR2180P</td>
<td>UGS0101P</td>
</tr>
<tr>
<td>UGM0055M</td>
<td>UGM0095P</td>
<td>UGN0220M</td>
<td>UGN0460T</td>
<td>UGR1070T</td>
<td>UGR1222P</td>
<td>UGR2180T</td>
<td>UGS1051M</td>
</tr>
<tr>
<td>UGM0055P</td>
<td>UGM0020M</td>
<td>UGN0220P</td>
<td>UGN0461P</td>
<td>UGR1071P</td>
<td>UGR1230M</td>
<td>UGR2181P</td>
<td>UGS1051P</td>
</tr>
<tr>
<td>UGM0055T</td>
<td>UGM0020P</td>
<td>UGN0220T</td>
<td>UGR1100P</td>
<td>UGR1080M</td>
<td>UGR1230T</td>
<td>UGR2182P</td>
<td>UGS1052P</td>
</tr>
<tr>
<td>UGM0056P</td>
<td>UGM0020T</td>
<td>UGN0221P</td>
<td>UGR1101P</td>
<td>UGR1080T</td>
<td>UGR1231P</td>
<td>UGR2190M</td>
<td>UGS1100M</td>
</tr>
<tr>
<td>UGM0060M</td>
<td>UGM0021P</td>
<td>UGN0240M</td>
<td>UGR1102P</td>
<td>UGR1081P</td>
<td>UGR1232P</td>
<td>UGR2190T</td>
<td>UGS1100T</td>
</tr>
<tr>
<td>UGM0060P</td>
<td>UGM0040M</td>
<td>UGN0240P</td>
<td>UGR1100P</td>
<td>UGR1082P</td>
<td>UGR1240M</td>
<td>UGR2191P</td>
<td>UGS1101P</td>
</tr>
<tr>
<td>UGM0060T</td>
<td>UGM0040P</td>
<td>UGN0240T</td>
<td>UGR1000P</td>
<td>UGR1090M</td>
<td>UGR1240T</td>
<td>UGR2192P</td>
<td>UGS1102P</td>
</tr>
<tr>
<td>UGM0061P</td>
<td>UGM0040T</td>
<td>UGN0241P</td>
<td>UGR1010P</td>
<td>UGR1090T</td>
<td>UGR1241P</td>
<td>UGR2200M</td>
<td>UGS1150M</td>
</tr>
<tr>
<td>UGM0062M</td>
<td>UGM0041P</td>
<td>UGN0280M</td>
<td>UGR1020P</td>
<td>UGR1091P</td>
<td>UGR1242P</td>
<td>UGR2200T</td>
<td>UGS1150T</td>
</tr>
<tr>
<td>UGS1151P</td>
<td>UGU0120P</td>
<td>UGU1141P</td>
<td>UGU3041P</td>
<td>UGZLXCHK</td>
<td>UGZ0029X</td>
<td>UGS0100M</td>
<td></td>
</tr>
<tr>
<td>UGS1152P</td>
<td>UGU0130P</td>
<td>UGU1142P</td>
<td>UGU3042P</td>
<td>UGZXXXXP</td>
<td>UGZ0030#</td>
<td>UGS1150T</td>
<td></td>
</tr>
<tr>
<td>UGS1160M</td>
<td>UGUX101T</td>
<td>UGU120UM</td>
<td>UGU3110M</td>
<td>UGZ0001P</td>
<td>UGZ0030P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1160T</td>
<td>UGUX110T</td>
<td>UGU130UM</td>
<td>UGU3110T</td>
<td>UGZ0002P</td>
<td>UGZ0032M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1161P</td>
<td>UGUX120T</td>
<td>UGU2010M</td>
<td>UGU3111P</td>
<td>UGZ0003P</td>
<td>UGZ0032P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1162P</td>
<td>UGUX130T</td>
<td>UGU2010T</td>
<td>UGU3112P</td>
<td>UGZ0004P</td>
<td>UGZ0034P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1170M</td>
<td>UGU101UM</td>
<td>UGU2011P</td>
<td>UGU3120M</td>
<td>UGZ0005P</td>
<td>UGZ0035P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1170T</td>
<td>UGU1010M</td>
<td>UGU2012P</td>
<td>UGU3120T</td>
<td>UGZ0006#</td>
<td>UGZ0036P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1171P</td>
<td>UGU1010T</td>
<td>UGU2020M</td>
<td>UGU3121P</td>
<td>UGZ0006M</td>
<td>UGZ0037P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1172P</td>
<td>UGU1011P</td>
<td>UGU2020T</td>
<td>UGU3122P</td>
<td>UGZ0006P</td>
<td>UGZ0038P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1190M</td>
<td>UGU1012P</td>
<td>UGU2021P</td>
<td>UGU4020M</td>
<td>UGZ0007P</td>
<td>UGZ0039P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1190T</td>
<td>UGU1020M</td>
<td>UGU2022P</td>
<td>UGU4020T</td>
<td>UGZ0008P</td>
<td>UGZ0040P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1191P</td>
<td>UGU1020T</td>
<td>UGU2040M</td>
<td>UGU4021P</td>
<td>UGZ0009P</td>
<td>UGZ0060P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1192P</td>
<td>UGU1021P</td>
<td>UGU2040T</td>
<td>UGU4022P</td>
<td>UGZ0010P</td>
<td>UGZ1001P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1200M</td>
<td>UGU1022P</td>
<td>UGU2041P</td>
<td>UGVERIFY</td>
<td>UGZ0011P</td>
<td>UGZ1002P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1200T</td>
<td>UGU1030M</td>
<td>UGU2042P</td>
<td>UGV0010M</td>
<td>UGZ0012P</td>
<td>UGZ9000P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1201P</td>
<td>UGU1030T</td>
<td>UGU2110M</td>
<td>UGV0010P</td>
<td>UGZ0013P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS1202P</td>
<td>UGU1031P</td>
<td>UGU2110T</td>
<td>UGV0010T</td>
<td>UGZ0014M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2051M</td>
<td>UGU1032P</td>
<td>UGU2111P</td>
<td>UGV0011P</td>
<td>UGZ0014P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2051P</td>
<td>UGU1040M</td>
<td>UGU2112P</td>
<td>UGV0030M</td>
<td>UGZ0015M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2052P</td>
<td>UGU1040T</td>
<td>UGU2120M</td>
<td>UGV0030P</td>
<td>UGZ0015P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2150M</td>
<td>UGU1041P</td>
<td>UGU2120T</td>
<td>UGV0030T</td>
<td>UGZ0016P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2150T</td>
<td>UGU1042P</td>
<td>UGU2121P</td>
<td>UGV0031P</td>
<td>UGZ0017P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2151P</td>
<td>UGU110UM</td>
<td>UGU2122P</td>
<td>UGW1110M</td>
<td>UGZ0018P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS2152P</td>
<td>UGU1110M</td>
<td>UGU2130M</td>
<td>UGW1110T</td>
<td>UGZ0019P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3051M</td>
<td>UGU1110T</td>
<td>UGU2130T</td>
<td>UGW1111P</td>
<td>UGZ0020$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3051P</td>
<td>UGU1111P</td>
<td>UGU2131P</td>
<td>UGW1112P</td>
<td>UGZ0020P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3052P</td>
<td>UGU1112P</td>
<td>UGU2132P</td>
<td>UGW1130M</td>
<td>UGZ0022P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3150M</td>
<td>UGU1120M</td>
<td>UGU3010M</td>
<td>UGW1130T</td>
<td>UGZ0023P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3150T</td>
<td>UGU1120T</td>
<td>UGU3010T</td>
<td>UGW1131P</td>
<td>UGZ0024$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3151P</td>
<td>UGU1121P</td>
<td>UGU3011P</td>
<td>UGW1132P</td>
<td>UGZ0024P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGS3152P</td>
<td>UGU1122P</td>
<td>UGU3012P</td>
<td>UGW1140M</td>
<td>UGZ0025#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGUX101P</td>
<td>UGU1130M</td>
<td>UGU3020M</td>
<td>UGW1140T</td>
<td>UGZ0025@</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
UGUI110P  UGU1130T  UGU3020T  UGW1141P  UGZ0025P
UGUI120P  UGU1131P  UGU3021P  UGW1142P  UGZ0026P
UGUI130P  UGU1132P  UGU3022P  UGZCLERP  UGZ0028#
UGUO101P  UGU1140M  UGU3040M  UGZINITP  UGZ0028P
UGUO110P  UGU1140T  UGU3040T  UGZLOADP  UGZ0029P
Chapter Nine - UMS Screen Mapping Procedures

Because the UMS System is CICS based, the use of Basic Mapping Support (BMS) is a virtual necessity. Several components must be produced. First is the MAP itself, an object module used at run time by CICS routines. Next is the storage definition of the data after being processed by BMS (on input or output). In UMS, a third DSECT, or storage definition is needed, the one for the programs working storage or communication area. There are some options on creating the first two items that will be mentioned here, but the third DSECT must be created by hand.

In this third definition, the data field must be UMS-compatible with the screen map definition; that is of the same type and length or of one that matches one of the LXTABLE’s automatic conversions such as PIC 9(5) into S9(4) comp. Each one of these fields must be preceded by two one-byte fields suffixed by “_Z” and “_T” respectively. These Z and T fields have functions in the LXTABLE processing cycle that are discussed in that chapter. If you do not plan to use the Z and T fields, that can be replaced by two filler bytes as PIC XX or DS CL2. For example, if you were operating with the field “DRIVER-NAME,” the DSECT fields might look as follows:

```
05 DRIVER-NAME-G.
10  DRIVER-NAME-Z          PIC X.
10  DRIVER-NAME-T          PIC X.
10  DRIVER-NAME            PIC X(25).
```

These T and Z field names are conveniently generated by the GFLD macro for assembler programs and the LX table assembly. The call `<fldname> GFLD `<asmpic>` is the pattern. For example:

```
DRVRNM GFLD CL25       will produce
```

```
   DRVRNM_Z DS  CL1
   DRVRNM_T DS  CL1
   DRVRNM   DS  CL25
```

The field DRVRNM is the one that will be mentioned in the LXTABLE entries. The LXTABLE processing assumes that the other two fields are represented. If they are not, the mapping results will be unpredictable.

The options come in the creation of the map itself and its DSECT. System development has access to the EDS Screen Generator program which, for licensing reasons, is not placed on the distribution tape. The generator is described later in this chapter.
The manual map generation process has been somewhat simplified. In place of coding the
DFHMSD and DFHMDI macros, code the required UMSHEADER macro. This macro
generates the DFHMSD and DFHMDI invocations along with a standard heading for all
UMS screens. Parameters on the macro provide the necessary customization as follows:

\[
\text{<name>} \text{ UMSHEADR } \text{<sfunc>},<\text{scrntitle}>,\text{TYPE=}<\text{type}>,\text{SYSTEM=}<\text{sys}>
\]

\[
\begin{align*}
\text{<name>} & \text{ is a required 1 to 6 character map name for} \\
\text{the assembly} & \\
\text{<sfunc>} & \text{ is a required positional parameter, a 1 to} \\
\text{4 character function code, served by this} & \\
\text{screen, that appears in the screen header} & \\
\text{<scrntitle>} & \text{ is a required positional parameter, a} \\
\text{1 to 80 character literal that will be a} & \\
\text{title appearing on the second line of the} & \\
\text{screen} & \\
\text{<type>} & = \text{NORMAL | DSECT the assembler output} \\
\text{form, Normal generates map statements, Dsect} & \\
\text{generates the DS definitions} & \\
\text{<sys>} & = \text{UMS | ALAR this determines the screen} \\
\text{date format: UMS = mm/dd/yyyy, ALAR =} & \\
\text{mm/dd/yy} & 
\end{align*}
\]

This macro generates 4 UMS-standard fields:

- The current date
- The time
- The function (from <sfunc>)
- A 50-character message line

These are all manipulated outside of the application program. To write messages, one only
specifies a message number (see chapter on Error Messages) and the proper text is inserted
by the system control programs.

Following this macro, code the standard DFHMDF macros to define each map field and
literal for the screen. The standard requires the names of the fields be of the form:

\[
\text{FLD<fnun> where <fnun> is a 3 or 4 digit number,} \\
\text{beginning with 5 that is zero filled, such as:} \\
\text{FLD005, FLD0007, FLD012}
\]
They begin with 5 as the UMSHEADR generates the first 4 fields in the map. After the last macro, one must supply the standard (CICS) `DFHMSD TYPE=FINAL` and an assembler `END` statement. When all these statements are assembled and linked, the map module is complete. The only problem with this process is the names of the fields. The mapping macros are assembler orientated and the names must be 7 characters or less, with the macro expansion supplying the eighth. Even though the newer assembler supports 30-character data names, the UMSHEADR macro does not currently support a language parameter and defaults to “LANG=ASM” in the DFHMSD macro. This forces one to hard code the COBOL storage definition. There are only two “tricks” to this.

♦ First, provide 12 bytes of filler at the head of the map storage.
♦ Second, provide three (3) bytes of storage ahead of each data field.

Using LXTABLE, one does not reference these fields so a definition of “PIC X(3).” is sufficient. Properly, they are first, “PIC S9(4) COMP.” followed by “PIC X.” This non-automatic description generation works well but one must be very careful that screen map changes affecting the length, type, or location of fields be reflected in two additional places, the hand-coded map “DSECT” and the storage definition for the application program which is accompanied by the “Z-field,” “T-field” components. The EDS map generator has the advantage of generating the map source (which must be modified) and its COBOL storage definition in the same pass. This will eliminate one source of potential errors. Remember, map/dsect misalignments produce “unpredictable” results. At the end of this chapter is the various “code” segments for a very simple map to illustrate the components required. The Map Generator discussion can be read to see these components from another perspective.

**Map Generator Description**

Basic Mapping Support (BMS) is used to define screens that interactive CICS on-line programs will display. Defining a screen means specifying all the fields on the screen and their attributes. The batch Map Generator program uses the screen definition (picture) to create a map macro and COBOL copy member of the screen variables.

In this description, the term “picture” is used to signify the TSO member that contains the necessary elements to define a screen (map). The first step in creating a map is to create the picture. Pictures of each UMS screen are stored by screen name in RMVMV.JMV.ISPPLIB. The UMS screen pictures also reside in RMVMV.RMV.UMSSRCEP and should be maintained in both libraries. If an SE creates a new UMS screen or modifies an existing one, he needs to promote the screen’s final picture to both libraries. If the picture for an existing screen is more current in one of the libraries than the other, the SE should use the more current one and put the final picture in both libraries. When the SE is creating a new UMS screen, he should use an
existing picture as a base to avoid having to key in all the parts of the picture member that are the same for all pictures.

At the top of the picture is the “control” section that includes the DFHMSD macro that is used to define the mapset. A mapset may contain one or more maps but the standard convention is one map per mapset. The DFHMSD macro specifies the mapset name (same as the map name) along with parameters that define the mapset. These parameters should never need to be changed.

Following the “control” section is the “screen” section. This section contains a representation of the screen which indicates exactly how the screen is to appear in the CICS region. The screen’s title is typed in along with all headers and field names. The length of each variable field is represented by the number of underline (“_”) marks. Five underline marks signify that the field may contain up to five characters. The Map Generator requires that variable fields be separated either by one space or by three or more spaces. The requirement exists so that each variable field may be preceded by its attribute byte and followed by a stopper byte. If only two spaces are provided between two variable fields, the Map Generator job will give a return code of four when executed. If one space is provided, it acts as both the stopper and the attribute byte.

Following the “screen” section is the “fields” section, which contains variable fields information (attributes) for the screen. This section is made up of columns, with headers specifying the various attributes. There is one screen field specified per row. To specify the attributes for a screen field, indicators are placed in the corresponding attribute columns. The screen fields are specified in the order that they appear on the screen (from left-to-right and top-to-bottom). The only field information required is the field name and its attributes. Below is a list of the most commonly specified attributes, the column header for each attribute, and the valid values that may be specified in each column:

- **PROT column**: indicates whether or not the operator may key data into the field. The field may be protected, unprotected or auto-skip, and these are specified by “P,” “U,” or “A” respectively.

- **NUM column**: indicates if the field is to contain numeric data only. This is specified by an “N” and is applicable only for terminals that have numeric lock capability.

- **DSPLY column**: indicates the field’s display intensity. This intensity may be normal, bright or dark (not displayed) and is specified by “N,” “B,” or “D” respectively.

- **FSET column**: indicates the field will always be treated as if the operator keyed data into it, whether he did or not. To indicate this, an “F” is specified in this column.
- CURSR column: indicates that the cursor will be placed at the beginning of this field when the screen is displayed. To indicate this, an “I” is specified in this column.

- JUSTFY column: indicates how data is to be justified in the field. Field justification can be left-justify/zero fill, left-justify/blank fill, right justify/zero fill and right-justify/blank fill. These justifications are indicated by “LZ,” “LB,” “RZ,” and “RB” respectively.

The map picture is completed once all the screen fields have been defined by indicating their names and attributes. After the picture has been created, the Map Generator job may be run to create the map macro and copy member.

The Map Generator JCL is located in RMVMV.JMV.USRJCL(T10JMAPG). It is to be used in conjunction with RMVMV.JMV.USRJCL(T10JMAPA), which is discussed below. The Map Generator will create a map macro and a COBOL copy member. These two created members will be named and placed in the libraries according to the SE’s specifications. To execute the Map Generator job, all the SE needs to do is retrieve T10JMAPG, change the picture name, change the DSLIB and BMSLIB library destinations and member names, and submit the job. A good EOJ is when all three proc steps have a return code of zero.

Once the map macro has been generated, the SE must make some changes to it so that it may then be assembled in the UMS environment. The map macro contains all the fields specified on the screen. The first four fields (Date, Time, Function, and MSG) on all UMS screens have their own map macro. This map macro is called “UMSHEADR” and resides in “RMVMV.RMV.UMSMACRO.” To make the changes to the new map macro, the SE needs to retrieve it and delete everything up to and including the definition of field four (which Map Generator has named “FLD0004”). Once this is completed, RMVMV.RMV.USRJCL(T10JMAPA) comes into play. This member contains instructions in a flower box and the JCL necessary to assemble maps (which will bring in the “UMSHEADR” member thus replacing all the code that was deleted). The SE will copy T10JMAPA into the top of his map macro (where the deleted code used to be). Once this is done, the SE needs to follow the instructions in the flower box. These instructions include changing the applicable map names and title, that were in T10JMAPA, to the correct names/title and then deleting the flower box. Once this is completed, the map macro member may be saved as the final version of the macro that will be moved to the production source libraries. This map macro member now contains the JCL necessary to assemble the map and can then be “subbed” to assemble it. A successful map assemble is indicated when the assemble and link-edit steps have return codes of zero.

Special notes:
The **UPIC** screen, which is used to attach vehicles to a policy, cannot use Map Generator. On this screen are three columns of vehicle information that contain data such as the vehicle’s VIN and registration (one column for each vehicle). The programs that perform the processing for the UPIC screen, require that the flow be left-to-right/top-to-bottom within each column. Map Generator takes a picture and creates a map macro with left-to-right/top-to-bottom for the whole screen and not for special columns. When the screen was created, the map macro fields were moved around so that the flow would be correct. If an SE needs to make a change to this screen, he/she will need to change both the picture and the map macro.

A word of caution: an SE may have made a change directly to a screen’s map macro and never updated the picture; therefore the picture may not reflect the current screen. The map macro is always current since its object code is what is executed. If an SE discovers this situation has occurred, he/she needs to update the picture with the previous changes along with his/her changes so it will be in sync for the next SE.

A final note is that an SE may have added some comments in the form of a flower box to the top of a map macro (between the JOB card and the EXEC card). This may contain information such as CSR number/date and the SE’s name. When the next SE makes changes to this screen, he/she needs to perform all the steps listed previously and copy this flower box into the updated map macro. By doing this, the flower box will not be lost when one moves the updated map macro to production.

### Map Generator Example

The following picture is an example of a simple map (MAPTEST) that would be stored in RMVMV.JMV.ISPPLIB. In this example, the row/columns for the screen fields may not match exactly between the picture and the map macro as this is a representation only. Additionally, the percent (%) sign is being used to represent the cent sign.

```
%CONTROL  MAPTEST
MAPTEST DFHMSD TYPE=DSECT,  X
CTRL=(FREEKB,FRSET),  X
LANG=ASM,
  X
MODE=INOUT,
TERM=3270-2,
TIOAPFX=YES,
STORAGE=AUTO

%SCREEN  MAPTEST
__________ _______ MASS REGISTRY OF MOTOR VEHICLES MAPTEST
THIS IS A MAP TEST SCREEN
FUNCTION: ___ MSG:  ____________________________________ PG __
LIC #: _________________________ ST: __

%FIELDS  MAPTEST
|----------------+-+--+----+---+---+---+-+-+-+-+-+-+-+-+-+-+-+-+
|                | |  |    |   |       |ATTRIBUTE  |  |J |H|V| |
|                | |L |    |   |       | | |D| | |C|C |U |I|A |
When the Map Generator job, RMVMV.JMV.USRJCL(T10JMAPG), runs, it takes the completed picture and creates a COBOL copy member and a BMS map macro. The created copy member would begin as follows:

```cobol
01 MAPTEST.
   05 FILLER PIC X(12).
   05 MAPTEST-DATE-G.
      10 MAPTEST-DATE-L PIC S9(4) COMP.
      10 MAPTEST-DATE-A PIC X.
      10 MAPTEST-DATE PIC X(10).
   05 MAPTEST-PASSWORD-G.
      10 MAPTEST-PASSWORD-L PIC S9(4) COMP.
      10 MAPTEST-PASSWORD-A PIC X.
      10 MAPTEST-PASSWORD PIC 9(05).
```

The rest of the screen’s fields in this copy member would be defined in much the same way. Each field has a group name and is broken down into three elementary data items: first is the length for the field, second is the attribute byte for the field, and third is the actual field. The password was given a NUMERIC attribute in the picture and thus received a “PIC 9” definition in the copy member.

The BMS map macro created by Map Generator would appear as follows (some of the macro code is omitted here in the interest of brevity):

```asm
MAPTEST DFHMSD X
   TYPE=DSECT, X
   CTRL=(FREEKB,FRSET), X
   LANG=ASM, X
   MODE=INOUT, X
   STORAGE=AUTO, X
   TERM=3270-2, X
```

When the Map Generator job, RMVMV.JMV.USRJCL(T10JMAPG), runs, it takes the completed picture and creates a COBOL copy member and a BMS map macro. The created copy member would begin as follows:

```cobol
01 MAPTEST.
   05 FILLER PIC X(12).
   05 MAPTEST-DATE-G.
      10 MAPTEST-DATE-L PIC S9(4) COMP.
      10 MAPTEST-DATE-A PIC X.
      10 MAPTEST-DATE PIC X(10).
   05 MAPTEST-PASSWORD-G.
      10 MAPTEST-PASSWORD-L PIC S9(4) COMP.
      10 MAPTEST-PASSWORD-A PIC X.
      10 MAPTEST-PASSWORD PIC 9(05).
```

The rest of the screen’s fields in this copy member would be defined in much the same way. Each field has a group name and is broken down into three elementary data items: first is the length for the field, second is the attribute byte for the field, and third is the actual field. The password was given a NUMERIC attribute in the picture and thus received a “PIC 9” definition in the copy member.

The BMS map macro created by Map Generator would appear as follows (some of the macro code is omitted here in the interest of brevity):

```asm
MAPTEST DFHMSD X
   TYPE=DSECT, X
   CTRL=(FREEKB,FRSET), X
   LANG=ASM, X
   MODE=INOUT, X
   STORAGE=AUTO, X
   TERM=3270-2, X
```
TIOAPFX=YES

MAPTEST DFHMDI

X

SIZE=(24,80)

FLD0001 DFHMDF POS=(001,002), LENGTH=0010,
JUSTIFY=(RIGHT,BLANK),
ATTRB=(ASKIP)

(FLD0002 through FLD0003 are omitted)

FLD0004 DFHMDF POS=(003,022), LENGTH=0050,
JUSTIFY=(LEFT,BLANK),
ATTRB=(PROT,BRT)
To complete the map macro, the programmer deletes all code from the top of the map macro through FLD0004 and copies RMVMV.JMV.USRJCL(T10JMAPA) in its place. Once this is completed, the contents of T10JMAPA are modified to reflect this map’s name/title. Following is the completed BMS map macro member:

```
//T10JMAPA JOB (5CGWBX,A51B),'MAPTEST', +
//              CLASS=S,MSGCLASS=V,NOTIFY=T07JSEB +
//STEP010 EXEC CACASMM,UMEM=MAPTEST +
//ASM.SYSIN DD *
TITLE 'MAPTEST -- THIS IS A MAP TEST SCREEN' +
MAPTEST UMSHEADR MAPT,'THIS IS A MAP TEST SCREEN' ++++++
   DFHMDF POS=(003,073),LENGTH=0002, X
   INITIAL='PG'
FLD0005 DFHMDF POS=(003,076),LENGTH=0002, X
   JUSTIFY=(LEFT,BLANK), X
   ATTRB=(PROT,NUM)
   (FLD0006 through FLD0007 are omitted)
DFHMSD TYPE=FINAL
END
```

The contents of T10JMAPA are indicated by the “+” bracket. In the sixth line of the map macro above, the value “MAPT” is the screen’s function code.

When the programmer submits this job, the macro will be assembled and will create the object code that is used for displaying the screen.
10

UMS Online Error Messages

The error-code passed with the outbound message text is structured as a 4 byte field, which is treated as a 1 word binary number. From the application perspective, it needs to be considered as a signed 9 digit numeric field.

<table>
<thead>
<tr>
<th>Digits</th>
<th>Length</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>2</td>
<td>major process identification</td>
</tr>
<tr>
<td>2-5</td>
<td>2</td>
<td>field grouping code - characters 4 through 7 of originatory program. i.e., for UGB0710P, these four characters would be ‘0710.’</td>
</tr>
<tr>
<td>6-8</td>
<td>3</td>
<td>sequence number</td>
</tr>
</tbody>
</table>

The concept is that a given code will be issued by **one module under 1 set of conditions**, only. The first 3 digits will relate to the module. The field grouping code will, within the module, attempt to localize the field(s) being processed when the error was detected. The detail code will attempt to further qualify the problem. These sub-field usages are designed to make analysis simpler on all sides. However, the general concept will be that a unique 9 digit code yields a unique message (from the message dictionary on the guest side). The only time the guest will be concerned with the sub-code breakdown is if its dictionary is out of sync and the text cannot be located.

The dictionary will contain a flag set for each message which indicate if the 8 byte error qualifier field is used for this code, and if so how. This structure is not defined at this time.

**Note:** If the same error-text is produced from more than 1 place, it is assigned multiple codes so that the source can be localized as required.

The Host Function/Dispatch/Security process reserves all Error Codes with digits 0-2 in the range of 100-199 for itself. Guest processes need to be aware of this value range.

UMS online programs move error numbers to the common area when an error is encountered. The Guest Side Cleanup module (UGZ0005P) matches the error number with an error number in the error message table and finds a corresponding error message. Error messages reside in a member named UGZ0004P in the UMS source libraries.
This member consists of list the following messages:

<table>
<thead>
<tr>
<th>Type of messages</th>
<th>Range of numbers</th>
<th>Copybook</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMS control CICS programs messages</td>
<td>199-200</td>
<td></td>
</tr>
<tr>
<td>UMS error messages</td>
<td>201-299</td>
<td></td>
</tr>
<tr>
<td>EDS messages</td>
<td>300-599</td>
<td>$EDSMSGS</td>
</tr>
<tr>
<td>MRB messages</td>
<td>600-699</td>
<td>$MRBMSGS</td>
</tr>
<tr>
<td>Stopper/enabler messages</td>
<td>800-899</td>
<td></td>
</tr>
<tr>
<td>Alars error messages and NDR responses</td>
<td>900-999</td>
<td></td>
</tr>
</tbody>
</table>

The EDS and MRB message copy members reside in the UMS macro libraries.

To change a message in $EDSMSGS or $MRBMSGS, the SE should copy the error message member from RMVM.RMV.UMSMACR3 library to RMVMV.RMV.UMSMACRO library, make the needed changes, copy the error message table UGZ0004P from RMVM.RMV.UMSSRCE3 to RMVMV.RMV.UMSSRCE, reassemble this table with new changes, move modified copybook and error message table (UGZ0004P) to CV14 and CV11 library using automated UMS move method.

To change a message (UMS, Alars, or Stopper/enabler message), the programmer should copy the error table UGZ0004P from RMVMV.RMV.UMSSRCE library to RMVMV.RMV.UMSSRCE, make the needed changes, reassemble the error table, move modified UGZ0004P (error message table) to CV14 and CV11 library using automated UMS move method.
Adding an Error Message to a Program

The members $EDSMSGS, $MRBMSGS, and UGZ0004P (error message table) are arranged by error number and program name. To add an error number, the programmer should find the sequence of numbers that exists for the program, and choose the next available number. If an error message is being added that already exists for a different program, then an alias is used as a pointer to the previous message. An alias can only be used if the new error number is greater than the alias number.

The programmer should copy $EDSMSGS or $MRBMSGS to RMVMV.RMV.MACRO (test library). If EDS message or MRB message must be added, then copy the error message table UGZ0004P from RMVMV.RMV.MACR3 to RMVM.RMV.MACRO, add new message and reassemble the error table.

Example:

The program UGRO011P needs an error message for "F8 INVALID, IMPROPER CURSOR POSITIONING". Following is a sample of the member $EDSMSGS:

```
ERR  203001010,UGRO010P,,'F8 INVALID, IMPROPER CURSOR POSITIONING'
ERR  203001011,UGRO010P,,'F8 INVALID, END OF SET ENCOUNTERED'
ERR  204001010,UGRO011P,,'ALIAS=200001011'
```

In this case, the new entry would appear as follows:

```
ERR  204001010,UGRO011P,,'ALIAS=203001010'
```

In this example the ALIAS points to the error message used for error 203001010.

UGZ0004P - The Message Module

UGZ0004P is a module also known as the message text module. This module is the central repository for all UMS system messages. The system programs use a message code passed in the commarea to the clean-up control module (UGZ0005P) which will translate the code into the text which is then put into the appropriate area on the screen.
Messages must be in ascending order by error-code.

ERROR DICTIONARY ELEMENT DEFINITION

&P0  ERR &P1,&P2,&P3,&P4

&P1 = Required 9 decimal digit error-code

&P2 = Required 1 to 8 Byte Program-name

&P3 = Optional 1 to 8 hex digit flagset

&P4 = Required 1 to 50 byte message

NOTE: The message can be replaced by text of the form of 'ALIAS=123456789'
where 123456789 is the code number of a previously defined message to
reuse the same text with a different error number.

Some Examples:

ERR  203001011,UGR0010P,,'F8 INVALID. END OF SET ENCOUNTERED.'
ERR  203001012,UGR0011P,,'ALIAS=200001011'
ERR  203001013,UGR0011P,,'ALIAS=200001011'
ERR  203001014,UGR0011P,,'ALIAS=201001013'
Assembled Example of a Message Module

Note: These examples have been slightly modified by text-edit to allow them to fit in the page or to improve their readability. For brevity, some macros have a number of parameters to show the results of expansion. Some combinations, while assembled correctly may be illogical. Please consult the detailed write-up for parameter usage.

EXTERNAL SYMBOL DICTIONARY

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>ID</th>
<th>ADDR</th>
<th>LENGTH</th>
<th>LD</th>
<th>ID</th>
<th>FLAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSECT1</td>
<td>SD</td>
<td>0001</td>
<td>000000</td>
<td>000064</td>
<td></td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CSECT2</td>
<td>SD</td>
<td>0002</td>
<td>000068</td>
<td>00006B</td>
<td></td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

ECE0004P - TEST GUEST-SIDE ERROR MESSAGE DICTIONARY

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJECT CODE</th>
<th>ADDR1</th>
<th>ADDR2</th>
<th>STMT</th>
<th>SOURCE STATEMENT</th>
<th>ASM H V 02 14.58 07/10/91</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PUSH PRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00110000</td>
</tr>
<tr>
<td>3</td>
<td>PRINT OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00120000</td>
</tr>
<tr>
<td>136</td>
<td>POP PRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01450000</td>
</tr>
<tr>
<td>137</td>
<td>GBLC &amp;OPTPR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01460000</td>
</tr>
<tr>
<td>138</td>
<td>&amp;OPTPR SETC '&quot;SYSParm'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01470000</td>
</tr>
<tr>
<td>139</td>
<td>AIF ('&amp;OPTPR' EQ 'GEN').TP020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01480000</td>
</tr>
<tr>
<td>140</td>
<td>AIF ('&amp;OPTPR' EQ '').TP020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01490000</td>
</tr>
<tr>
<td>141</td>
<td>AIF ('&amp;OPTPR' EQ 'NOGEN').TP010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01500000</td>
</tr>
<tr>
<td>142</td>
<td>TP010 ANOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01520000</td>
</tr>
<tr>
<td>143</td>
<td>PRINT GEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01530012</td>
</tr>
<tr>
<td>144</td>
<td>TP020 ANOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01540000</td>
</tr>
<tr>
<td>000000</td>
<td>CSECT1 START 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01550000</td>
</tr>
</tbody>
</table>

Chapter Ten - UMS Online Error Messages
### Registry of Motor Vehicles – UMS Programmer’s Manual

#### Chapter Ten - UMS Online Error Messages

146 * START OF TABLE PREFIX 01560000

**000000 00000000**

147 NUMENT DC F'0'

**000004 0000002C**

148 DC A (FIRSTTAG)

**000008 00000068**

149 FIRSTDTL DC A (CSECT2)

**00000C 00000100**

150 DC A (RELEASE)

151 RELEASE DC AL2 (030), CL9 'SYSDATE'

DC CL5 'SYSTIME', C 'VER 03.01=' 01600000

**000010 001EF0F761F1F061**

152 FIRSTTAG DS 0F

153 * END OF TABLE PREFIX 01630000

**00002C**

154 ERR 100000001, UHZ0001P, 'HOST MODULE NOT FOUND'

**00002C 05F5E101**

155+CSECT1 CSECT

01-00125

**000030 14000000**

156+D100000001 DC F'100000001'

+ DC AL1 (L'TG30001-1+X'0'), AL3 (TG20001-CSECT2)

000068

157+CSECT2 CSECT

00-00127

000068 C8D6E2E340D4D6C4

158+TG200001 EQU *

00-00128

000068

159+TG300001 DC C 'HOST MODULE NOT FOUND'

00-00132

000034

160+CSECT1 CSECT

00-00133

161 ERR 100000002, UHZ0022P, 'UNKNOWN HOST FUNCTION'

000034

162+CSECT1 CSECT

00-00125

000034 05F5E102

163+D100000002 DC F'100000002'

+ DC AL1 (L'TG30002-1+X'0'), AL3 (TG20002-CSECT2)

00007D

164+CSECT2 CSECT

00-00127

00007D

165+TG200002 EQU *

00-00128

00007D E4D5D2D5D6E6D540

166+TG300002 DC C 'UNKNOWN HOST FUNCTION'

00-00132

00003C

167+CSECT1 CSECT

00-00133
Chapter Ten - UMS Online Error Messages

168 ERR 100000003,UHZ0333P,,'RECURSIVE DUP-KEY CALL'
00003C 169+CSECT1 CSECT 01-00125
00003C 05F5E103 170+D100000003 DC F'100000003'
000040 1500002A + DC AL1(L'TG30003-1+X'0'),AL3(TG20003-CSECT2)
000092 171+CSECT2 CSECT 01-00127
00092 D9C5C3E4D9E2C9E5 172+TG20003 EQU * 01-00128
000092 173+TG30003 DC C'RECURSIVE DUP-KEY CALL' 01-00132
000044 174+CSECT1 CSECT 01-00133

175 ERR 456789004,UHZ0001P,,'DUP-KEY NOT SPECIFIED'
000044 176+CSECT1 CSECT 01-00125
000044 1B3A0C0C 177+D456789004 DC F'456789004'
000048 14000040 + DC AL1(L'TG30004-1+X'0'),AL3(TG20004-CSECT2)
0000A8 178+CSECT2 CSECT 01-00127
0000A8 C4E4D760D2C5E840 179+TG20004 EQU * 01-00128
0000A8 180+TG30004 DC C'DUP-KEY NOT SPECIFIED' 01-00132
00004C 181+CSECT1 CSECT 01-00133

182 ERR 567891123,UHZ0001P,,'ALIAS=100000003' 01680012
00004C 183+CSECT1 CSECT 01-00118
00004C 21D954B3 184+A567891123 DC F'567891123'
000050 FF00003C + DC XL1'FF',AL3(D100000003-CSECT1) 01-00119

185 ERR 776721234,UHZ0003P,12345678,'ALIAS=456789004'
000054 186+CSECT1 CSECT 01-00118
000054 2E4BD352 187+A776721234 DC F'776721234'
000058 FF000044 + DC XL1'FF',AL3(D456789004-CSECT1) 01-00119

188 ERR 880000102,UHZ0003P,FFEE1234,'INVALID MVE LENGTH'
00005C 189+CSECT1 CSECT 01-00125
Chapter Ten - UMS Online Error Messages

MVS/XA DFP VER 2 LINKAGE EDITOR 14:58:28 WED JUL 10, 1991
JOB RMCJEBBP STEP STEP010 PROCEDURE LKED
INVOCATION PARAMETERS - LIST,XREF
ACTUAL SIZE=(317440,79872)
OUTPUT DATA SET RMVMV.UMS.RMCJEBB IS ON VOLUME RELP02

CROSS REFERENCE TABLE

CONTROL SECTION          ENTRY
NAME   ORIGIN  LENGTH   NAME   LOCATION   NAME   LOCATION   NAME   LOCATION

...
CSECT1 00  64
CSECT2 68  6B

LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
ENTRY ADDRESS  00

TOTAL LENGTH  D8
** ECE0004P REPLACED AND HAS AMODE 24
** LOAD MODULE HAS RMODE 24
** AUTHORIZATION CODE IS 0.
Hexadecimal dump of Message Module Example

AMASPZAP INSPECTS, MODIFIES, AND DUMPS CSECTS OR SPECIFIC DATA RECORDS ON DIRECT ACCESS STORAGE.

```
DUMPT ECE0004P ALL 00110006

**CCHHR- 0011000336 RECORD LENGTH- 0000D8 MEMBER NAME ECE0004P CSECT NAME CSECT1
000000 00000007 0000002C 00000068 00000010 001EF0F7 61F1F061 F9F140F1 F44BF5F8 ..................... *

*..07/10/91 14.58*

000020 7EE5C5D9 40F0F34B F0F17E00 05F5E101 14000000 05F5E102 14000015 05F5E103 *=VER 03.01=..5..* *

*......5.......5..*

000040 1500002A 1B3A0C0C 14000040 21D954B3 FF00003C 2E4BD352 FF000044 3473BC66 ........................... *

*.....L.........*

000060 95000055

**CCHHR- 0011000336 RECORD LENGTH- 0000D8 MEMBER NAME ECE0004P CSECT NAME CSECT2
000000 C8D6E2E3 40D4D6C4 E4D3C540 D5D6E340 C6D6E4D5 C4E4D5D2 D5D6E6D5 40C8D6E2 *=HOST MODULE NOT* *

*FOUNDUNKNOWN HOS*

000020 E340C6E4 D5C3E3C9 D6D5D9C5 C3E4D9E2 C9E5C540 C4E4D760 D2C5E840 C3C1D3D3 *=T FUNCTIONRECURS* *

*IVE DUP-KEY CALL*

000040 C4E4D760 D2C5E840 D5D6E340 E2D7C5C3 C9C6C9C5 C4FFEE12 34C9D5E5 C1D3C9C4 *=DUP-KEY NOT SPEC* *

*IFIED....INVALID*

000060 40D4E5C5 40D3C5D5 C7E3C8 *

MVE LENGTH*

AMA113I COMPLETED DUMP REQUIREMENTS

AMA100I AMASPZAP PROCESSING COMPLETED

****************************************************************************** BOTTOM OF DATA ********************************************************************************
11 Special System Functions

This chapter is essentially a continuation of Chapter 2, Control Dispatch. The system functions described here are more specialized in their operation and were therefore separated from the “main line” CD discussion.

Limited Secondary Session (LSS)

This service allows cursor selection of functions from within other functions, which are not scroll functions, the execution of that function, and the return to the original function at the exact point of exit. This is similar to sub-routine execution. The service works in conjunction with the PA1 secondary-session facility. The LSS session differs from the PA1 facility in that, when invoked, a function-name and surrogate are passed to the session. In this mode, the user cannot change the function (prohibited by Control Dispatch) and the only function keys allowed are 7, 8, 9, and 12. All other function keys are treated as a “return to ordinary session” request. The only function changes which can occur are those accomplished by internal dispatch. The user is allowed to change key fields for the invoked function.

The concept of “reference list” is introduced. This list is very similar to the scroll table but is located in other storage media. It is intended to provide a cursor selection entry to the LSS which is independent of the scroll facility and is a detailed reference mechanism to it.

As with F6 (screen hop), any function which can receive a F4 selection is eligible to receive this service and cannot differentiate it from a F4 entry. F3 is used for cursor selection from a reference-list, and can only be used from their primary session. Any program is eligible to build a reference list and no PCTEG entry is required for this eligibility. Whenever a function change occurs on the guest (even if it occurs via internal dispatch or host function swap), any existing reference-list is purged. A module was built to handle the reference-list. Its name is stored in guest common in the eight bytes following the name of the internal dispatch module.

The reference list itself is built in the end of the map-output area. This should require no special changes as LXTABLE manages these areas. If the program does not use the LXTABLE mechanism for mapping it will have to take care in the use of the storage. After a link to the reference list manager the 572 bytes of the output map area are cleared by the manager. Applications must not use this space unless they call the manager. If they do use this space, it may confuse LXTBL or SENDBACK. A DSECT for the area is in Appendix B. If the reference list manager is called with entries-per-line equal to low-
values, the reference list will be purged. Each call to the manager sets a return code in
the guest common error-code. If it is non-zero, the application should exit and allow the
error to be posted. The manager is called by a CICS LINK passing guest common.
Because a LINK is involved and the manager stores the reference list on external media,
excess calls MUST be avoided.

**UMS Screen Hop Facility**

Screen hop is defined as a mechanism of transferring from one business function to
another where the logical business key being accessed in the first function presents the
external appearance of being automatically transferred to the second. In UMS, the
terminal operator will indicate a desire to accomplish such a transfer by means of F6. In
the context of such a transfer, the business function during which F6 is pressed is referred
as the “hop-from” function, and the new function keyed at the top of the screen is
referred to as the “hop-to” function.

From a technical perspective, the use of F6 will be transparent to the hop-to function. On
terminal operator’s use of F6 will be indistinguishable from use
of F4. The entry-reason code will contain the same value, and the F4/F9 surrogate
type/value will be setup in the same fashion. The 60 bytes in guest common called
UGSURLST, which was previously not referenced (but reserved for screen-hop usage)
and defined as 15F has undergone a definition change. It is now 12(CL1,FL4). This
causes it to still be 60 bytes long, but provides for surrogate-types as well as surrogate
values.

The hop-from function has some responsibility in the use of this 60 byte area, as well as
the control-dispatch software. In order to properly use the 60 byte area, all functions
must be categorized relative to this area. The possible categorizations are:

A. primary
B. supportive
C. neutral
D. primary & supportive

A function defined as primary puts definitive information in the table. It typically
operates from a definitive key. Examples of this type of function are ULI and URI. ULI
would place a PERS surrogate and type in the table. URI would place PERS, VEHR,
VEHT, VEHC, and VMOD surrogates and types in the table. Because these are primary
functions, they have sole responsibility for the table. Therefore, before going to the host
to read a record they must move low-values to the table. After a successful return from
the host, they must place the appropriate surrogate types and values in the table. The
order in which the entries are placed in the table is unimportant to the control - dispatch
functions, but does represent the order in which the entries are evaluated during an F6
transfer, and thus can be viewed as a priority definition mechanism.
A function defined as supportive adds information to the table. It is usually used to get more information about a primary key. The primary key would generally have been entered on a previous screen. An example of this in ALAR would be SP. SP typically operates from an existing VEHR or PERS key, and gets parking offense data (INCD-OFNS & ACTN surrogates). If this function were implemented in UMS, **before** going to the host with an existing key, it would SEARCH the table and if it found any type entries which were INCD-OFNS or ACTN it would put low-values in the type entry. **After** a successful return from the host, it would again search the table, and use the first entries with low-values in the surrogate-type for its INCD-OFNS and ACTN entries.

**Example:** SP, before going to host:

2222-SETUP-TABLE.
SET UGCOMMON-RESERVED-SURR-INDEX TO 1.
2222-SEARCH-TABLE.
SEARCH UGCOMMON-RESERVED-SURR-LIST
WHEN
UGCOMMON-RESERVED-SURR-TYPE(UGCOMMON-RESERVED-SURR-INDEX) = (incd-ofns-surr-type OR actn-surr-type)
MOVE LOW-VALUES TO
SET UGCOMMON-RESERVED-SURR-INDEX UP BY +1
GO TO 2222-SEARCH-TABLE.
2222-EXIT.
EXIT.

**Example:** SP after successful return from host:

3333-FIXUP-TABLE.
SET UGCOMMON-RESERVED-SURR-INDEX TO 1.
SEARCH UGCOMMON-RESERVED-SURR-LIST
AT END GO TO 3333-EXIT
WHEN
UGCOMMON-RESERVED-SURR-TYPE(UGCOMMON-RESERVED-SURR-INDEX) = LOW-VALUES
MOVE INCD-OFNS-SURR-TYPE TO
UGCOMMON-RESERVED-SURR-TYPE(UGCOMMON-RESERVED-SURR-INDEX)
MOVE INCD-OFNS-SURR-VALUE TO
UGCOMMON-RESERVED-SURR-VALUE(UGCOMMON-RESERVED-SURR-INDE X)
SET UGCOMMON-RESERVED-SURR-INDEX UP BY +1.
SEARCH UGCOMMON-RESERVED-SURR-LIST
AT END GO TO 3333-EXIT
A function defined as neutral does not change the table.

A function defined as primary and supportive may need both types of logic mentioned above. SP might be an example of this if it is entered by other than (F4 or F6) and a license# or regno key directly keyed.

The expression of the above categorizations is in the code contained within the applications functional processors. No UMS tables are changed for this support. The decisions relative to the categorizations are a business type decision. For UMS versions of existing ALAR applications, these decisions are made by duplicating existing ALAR functionality. For new applications, a specification level decision is required.

UMS control-dispatch will clear the above mentioned 60 bytes whenever a function-change occurs without the pressing of F6.

**ALARS-UMS Bridge**

The bridge is a generic name for several software components that allow the user semi-automatic transition between the two major application control/dispatch (supervisory) systems WITHOUT the user being required to log-off one and logon to the other. In this respect, the bridge is like a session-manager. Additionally, the bridge automatically performs some data passing functions between the systems and provides a semi-smooth transition between some of the differences across the systems. In particular, the bridge passes (in both directions) data for screen hopping and in this area provides functionality dependent on an intimate knowledge of both applications and beyond that provided by session management software.

The bridge assumes that a user will log on to ALAR. Thereafter, if they invoke a UMS function (such as LI or LX), an automatic transfer to UMS (and the selected function) will occur. At such time as the user selects (from the UMS side) an ALAR function (such as LS), an automatic transfer back to ALAR (and the selected function) will occur.
If the user logs directly onto UMS (does not come from ALAR) and issues an ALAR function, they will be forced to the ALAR signon screen, and any screen-hop data will be lost. In a similar fashion, if the user changes signon-id while at the UMS side, the same result will occur.

True UMS Guests can not get into ALAR via the bridge. The bridge is only functional if (1) the UMS guest is in HOST=GUEST mode (not a true LU6.2 GUEST) and (2) there is a real ALAR system in the same CICS region.

The bridge is bi-directional, but uses different software in each direction. From the ALAR side, there needs to be a normal MRMXDTBL entry for each function recognized as being processed on the UMS side. Each of these functions must point at MLIP090P. It makes no difference if the MRMXDTBL entry is setup as an update entry or not. All the normal security provisions must be present. Of course, the target function must exist on the UMS side. No changes are required to the UMS side.

Coding in MRMXDTBL and MRMP010C assumes that the function-name section of MRMXDTBL has a fixed maximum length and is then followed by the program-name section. Coding in UMS does not rely on the fixed maximum length, but does assume that there will be at least 1 unused entry (high-value key) in MRMXDTBL. This has no effect or consequence for the foreseeable future. However, at such time as enough additional functions might be added to MRMXDTBL to EXACTLY fill the function-name section, the bridge would cease to work (at this time, no more functions could be added to ALAR without recompilation of MRMP010C and a change to the base number of functions). Thus there is a difference of 1 between the point where MRMP010C needs to be modified because of the requirements of the bridge and the point where MRMP010C needs to be modified because of the requirements of its own functionality and limitations. In neither case does the UMS code need to be modified. Also, many functions need to be added to MRMXDTBL before this problem occurs.

At run time, if a UMS guest is running HOST=GUEST, AND there is an ALAR in the same CICS region, an alternate PCTEG table is dynamically built at guest startup to reflect the ALAR functions. When UMS detects a function-change, it searches first in the UMS PCTEG table, next in the function-alias table and finally in the dynamically generated ALAR PCTEG table. At startup time, the entries generated into the ALAR PCTEG table reflect each ALAR function which is not mapped to the ALAR side bridge program (MLIP090P). The combination of this action AND the UMS search order insure that a loop of switching between the systems will not occur. The UMS initialization code includes two tables. These tables are referred to as a SPECIFIC function table and a GENERIC function table. These two tables enable UMS to decide what types of surrogate entries to map into the dynamically generated PCTEG entries. If a function is found in the SPECIFIC table, the surrogate-type string contained there is propagated into the generated PCTEG entry. If not, the GENERIC table is searched for the first byte of the function. If it is found, the associated surrogate-type string is propagated into the generated PCTEG entry. In any other case, the function cannot carry screen-hop data from UMS to ALAR.
From ALARS to UMS, when MLIP090P is invoked, it must differentiate between a screen hop and a simple transfer occurs. To do this, it matches the selected function-name first to the UMS Guest's PCTEG and second to the UMS Guest's function-alias table. After getting a hit, it interrogates any PFC surrogate string inherent in the entry. If the PFC surrogate strings exist, any surrogates stored in the terminals ALAR common area are checked for being non-null and of a type specified in the PFC surrogate string. This checking is done in the priority order specified by the UMS surrogate string definition. If a match occurs, the appropriate surrogate value is passed to the UMS application just as if a user totally within UMS had accomplished a transfer between two UMS applications with PFC. This results in the bridge making the same judgement that ALAR applications make for screen-hop. It also causes PFC selection to work.

From UMS to ALAR, deciding if the call is for a screen hop or a simple transfer is much simpler than ALAR to UMS. In this case, the UMS system knows explicitly if a screen-hop is requested, and what kind of selection the target function accepts. Assuming that the SPECIFIC and GENERIC tables used at UMS startup are accurate reflections of the applications ability, the correct transfer will always occur, or an error message indicating that the user has requested an impossible transfer will occur.

The bridge’s function is invoked only at the time of transition between the two systems. While one is functioning in either system, the bridge is not operative. Thus, the user is subject to the rules of whichever system they are in at the moment. There are very few areas of difference. These differences should not present a problem.

**GHOST Program - UNI / NDR / CDLIS**

The external system interface, as used to communicate with NDR, is called the Unified Network Interface (UNI). It is an integration of the UNI/CDLIS product, the various CICS regions used at RMV, the ALAR product, and the UMS host/guest structure.

The design objectives of this software are:

1. To provide a common point of transmission to NDR
2. To provide a common point of receipt from NDR
3. To provide a mechanism of recovery from an NDR-down condition (or network-down) which is independent of manual interaction
4. To relieve the broad local application set from responsibility in the exact nuances of the UNI structure
5. To automatically relate the source of an inquiry (CICS/IDMS) to and from the associated UNI environment.

The Major Components are:

A. Routing Table - This is a table which relates the various CICS regions (by ACB name) with their associated IDMS regions and their host UNI region.

B. Transfer File - This is a VSAM dataset used to hold the record of an inquiry (currently only to NDR) and its eventual response. There is one transfer file for each associated IDMS region. The transfer file(s) are owned by the associated UNI region.

C. Send Program - This is a module which converts the applications intention to query NDR to a properly formatted inquiry, and routes this inquiry to the appropriate ESI region. It is responsible for obtaining the necessary data from the database, determining if test or production status is appropriate and for formatting the data according to the proper parameters. It runs in the CICS region initiating the interaction and uses remote resources in the UNI region.

D. Receipt Program - This is a module which converts an ESI response into transfer-file format, and adds it to the transfer-file. It also processes a variety of errors which might be reported by UNI. It runs only in the UNI region(s) and differentiates which transfer file the various responses might belong on. It performs no database access and does not depend on the originating application region being up.

E. LX - This is an application function written in UMS format which uses UMS host functions to exercise each of the above components/functions. It can invoke host functions which cause initiation of the send program (issue an NDR inquiry), display the results of the receipt program, create noncash records based on responses and delete responses.

F. The Ghost - This is the cleanup function. It is more completely described below.

This software is designed to view its environment as a set of concurrently executing independent processes with a small amount of overlap (or intersection). The processes are:

1. The originating application region
2. The associated UNI region
3. The target system as linked through UNI
In a practical example, an inquiry process initiated through LX would involve the following steps:

<table>
<thead>
<tr>
<th>GUEST</th>
<th>HOST</th>
<th>UNI</th>
<th>NDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>initiate request</td>
<td>send to UNI</td>
<td>send to NDR</td>
<td>execute rqst</td>
</tr>
</tbody>
</table>

Request complete ----> reformat & store

_____ UMS-NDR Query ______| UMS-NDR 'Reply'

request display data
retrieve data from UNI
return data

display data

Please note that the above processes are asynchronous. That is, once the initiation sequence is complete, the guest operator is free to perform any other function (LX or other), and return to examining the response at some time in the future (an hour, a day). There may be no response before the end of the day. The operator can, if they so desire, go to LX to await a response which should normally return in a minute or so.

Since these processes are not connected synchronously, and in reality cannot be unless all processing is to be suspended if some remote component is down, there needs to be some mechanism which deals with the various problems which will arise when one of the following occurs:

♦ All does not go perfectly
♦ There is no operator waiting to take action for the response on LX
♦ The transfer-file needs to be maintained

The mechanism providing these services is known as the ghost (program, UHL0020P).

The ghost program exists in CICS regions and there is one for each associated IDMS database. In cases where more than one CICS is connected to a single IDMS, the ghost runs in only one of the CICS regions: the one with the fewest service hours. For example, the CV11 ghost runs in the PROD region.
The GHOST is configured as a UMS Guest. It is free running time-initiated software which signs itself on to a UMS host (just as an insurance company would) and issues UMS host functions. It is a unique guest (perhaps very similar in concept to ISI’s software) since it does NOT use the standard guest packaging, but only uses the standard guest interface program to communicate with its host.

The ghost is configured to run automatically at intervals throughout the day. When it runs, it examines records on the transfer-file and uses the same host functions used by LX to take various actions:

1. If an inquiry to NDR has been sitting without a response for an unacceptably long interval AND there is evidence that NDR is (now) active, the inquiry is resent.

2. If a response from NDR has been sitting without action for a defined interval (where the interval is dependent on source and content), it is automatically applied.

3. If an applied response from NDR has been sitting for a defined interval (45 days) it is deleted.

When the ghost program runs it produces a report of activity as an audit trail.
12 UMS Batch

Tape Processing

Insurances companies send tapes (reels or cartridges) to the Registry. These tapes contain UMS transactions to be processed against the Registry’s database. Tapes received must have either UMSTEST or UMSTRNS as the dataset names. UMSTYEST datasets will be run against the CV14 database while UMSTRNS datasets will be run against the CV11 database. The following steps document the handling of UMS tapes.

1. The insurance companies will send tapes to production control. The control log for each tape should be verified with the tapes received.

2. After the control log has been verified, each tape should be separately run through the tape verification job. This job can be run anytime during the day to verify tapes to be processed that night. The purpose of the verification job is to screen the tapes before they are input to the clist. This will prevent abends from occurring later in the cycle. The volser, tape type (tape or cart), and ICO number must be entered into the JCL.

   The jobs can be found in MSTRJCL. These jobs will verify the data-set name, ICO number, and transaction count. These jobs will only process one tape at a time and must be run for each tape prior to input into the clist for batch processing.

3. Upon execution of the tape verification job, the sysout in the hold-queue must be checked. If the return code is “00” the transaction count from the job should be checked against the count on the corresponding control log. If the counts match, the tape can be input into the clist. If the counts do not match, the submitting insurance company is to be contacted about the discrepancy. Listed below are abends and the proper action to take.
Abend | Description | Action
--- | --- | ---
S813 | Invalid Dataset name | Contact the ICO submitting tape
2090 | Missing JCL Parm | JCL parm must be present
2120 | Invalid JCL parm length | JCL parm must be 3 characters long
2130 | Mismatch on ICO | Validate ICO parm with control log

If the ICO on the control log does not match the ICO on the tape, the submitting insurance company should be contacted and the tape returned.

4. After the UMS tapes have been verified, they may be entered into the CLIST. The CLIST is accessed through the “master Application Menu” in TSO. From there either the production or CV14 CLIST is selected.

   - A security screen will appear. The password “STING” should be entered.

   - The UMS Tape Log screen will be displayed.

   - UMS tapes are logged by entering the requested information. The VOLSER out field is optional and, if not entered, will default to the input VOLSER.

   - After logging all the input tapes to be processed, enter an “X” in the process field at the top of the screen to exit the clist panel.

   - A message indicating the total number of tapes logged will display. If the message does not display, an error has occurred. The input process should be repeated if an error occurs.

5. Updating the UMS cycle window is critical to controlling the number of tapes and transactions processed in the cycle. This clist is also accessed from the “Master Application Menu.” The appropriate batch window is selected (production or CV14).

   - The UMS cycle window screen will be displayed.

   - Enter the time available for the UMS cycle to run in minutes.

   - Enter “X” in the process field to exit the clist panel.
- A message indicating the total number of transactions to be processed will be displayed. If the message does not display, there is an error and the entire process should be repeated.

6. If tapes are to be processed the jobs need to be run are B000, B010, and B020. Monthly jobs must be run after B000 and B020.
13 Record Surrogates

The database is, for the most part, a standard IDMS database. It is described, with Bachman diagrams, in a separate document. These documents are maintained by DBAs. The most unusual feature is the use of the ‘surrogate records.’ There are six surrogate record types listed below. They are stored in single record-type areas with a key which is a full-word binary number through the CAL-S algorithm, rather than the usual IDMASCALC. The next surrogate number comes from a single instance of one record-type, whose value is incremented and then replaced. The algorithm uses the surrogate number essentially as the records ordinal position within the area. It can, with some information about the area such as page size and SMI interval, derive the page it should be stored on. The record goes in the calc chain for that page. If the area has only one record type, the suggested page is always suitable and there is no overflow at all.

The beauty of this arrangement is that the area may be page-expanded, have pages added or be unloaded and reloaded into new file extents with a new geometry. In any case, the CAL-S algorithm will function undisturbed as the nth record in the area. It will always be the nth, no matter what changes are made. This surrogate number, which is a unique key for each of its record types, can be (and is) used as a calc-key for records, in other areas, stored by IDMASCALC. The uniqueness and uniform size of these surrogate keys have provided very good space-management results in the other areas. The surrogate values are also stored as foreign keys in other record types. As with any foreign key, the target (surrogate) record type is implied by its use.

The conversion from various sorts of keys, such as SSAN or vehicle registration number, to a related surrogate is often made by the host software. In such a case, the host controller, using the transaction’s PCTEH entry, invokes the first named module to convert the ‘key value’ to a surrogate. If none is found, an error is returned. If one is found, control goes to the second, and succeeding (if any), module(s) named in the PCTEH entry to complete the business function processing. If two or more ‘matches’ are found, the duplicate transaction specified is invoked to switch function and return the first screen of duplicates.

On the applications side of the system, there are requirements to manipulate lists of surrogate keys for multiple record types. To differentiate them, a one-byte prefix is defined for the six types used and then prefixed to the 4-byte surrogate key. The six types and their values are listed below. The following few lines from the copy book UGZCOMMC show the structure in which they are placed.
10 UGCOMMON-RESERVED-SURROGATES.
15 UGCOMMON-RESERVED-SURR-LIST OCCURS 12 TIMES INDEXED BY
      UGCOMMON-RESERVED-SURR-INDEX.
20 UGCOMMON-RESERVED-SURR-TYPE        PIC X.
20 UGCOMMON-RESERVED-SURR-VALUE    PIC S9(9) COMP.

The order for the record types (which may only be for a single type) is dictated for F4 and
F9 by the 1 to 4 type codes in the fields UGCOMMON-F4-STRING and UGCOMMON-
F9-STRING. These values are set by the System Control Program during initialization
from the values in the corresponding strings located in the PCTEG entry. The code
segment for these fields follows:

10 UGCOMMON-CUR-FUNC-CTRL-FLAGS.
15 UGCOMMON-PROCESS-BYTE-1       PIC X.
15 UGCOMMON-PROCESS-BYTE-2       PIC X.
15 UGCOMMON-PROCESS-BYTE-3       PIC X.
15 UGCOMMON-PROCESS-BYTE-4       PIC X.
15 UGCOMMON-USER-FLAGS.
20 UGCOMMON-PF4-STRING    PIC X(04).
20 UGCOMMON-PF9-STRING   PIC X(04).

For scrolling, there is a different table that will hold 140 surrogate keys only. The type
identification in this case is made from the 1 to 8 type codes stored in UGCOMMON-
SCROLL-SURR-TYPE-TBL. Whatever sequence is here is assumed to repeat enough
times to cover the number of surrogate keys shown by the index. The fields involved are
as follows:

05 UGCOMMON-SCROLL-BASE-DATA             PIC X(742).
05 FILLER REDEFINES UGCOMMON-SCROLL-BASE-DATA.
10 UGCOMMON-LAST-SCROLL-FUNC      PIC X(04).
10 FILLER                         PIC X(02).
10 UGCOMMON-SCROLL-FORMAT        PIC X(12).
10 FILLER REDEFINES UGCOMMON-SCROLL-FORMAT.
15 UGCOMMON-SCROLL-DATA-LINE1    PIC S9(4) COMP.
15 UGCOMMON-SCROLL-SURR-TYPE-TBL PIC X(08).

NOTE: Intervening fields are omitted to save space. Refer to Appendix F
for a complete listing.
10 UGCOMMON-SCROLL-TABLE PIC X(560).
10 FILLER REDEFINES UGCOMMON-SCROLL-TABLE.
15 UGCOMMON-SURR-NUMB OCCURS 140 TIMES
   INDEXED BY UGC-SURR-INDEX
   PIC S9(09) COMP.
05 UGCOMMON-APPLICATION-WORK-AREA PIC X(512).

The 20 Types of ‘Application’ Surrogates defined are:

0 = VEHR [S] (registration)          1 = VEHC [S] (veh. claim)
2 = VEHT [S] (veh. title)            3 = VMOD [I] (veh. model)
4 = PERS [S] (person record)         5 = NOWN [S] (corp. owner)
6 = PLCY [S]                          7 = OACD [S]
8 = OVID [S]                          9 = OWTH [S]
A/B = CITA/IVIO [S]                   C = ACTN [S]
D = DECI [S]                          E = PINQ [S]
F = AUTI [S]                          G = SUBM [S]
H = SDIP XCLU [S]                     M = REIN EXPA [S]
V = MSCD [S]                          W = DISP [S]

Low-Values or Spaces = None

NOTE: [S] surrogate is key for record stored CAL-S
      [I]                                       IDMSCALLC

NOTE: Additional mention of surrogates is found in the screen-hop function
discussion on page 182.
Appendices
Appendix A:  

Non-UMS LXTABLE Edit Example

NON-UMS LXTABLE DISCUSSION

In its native form, the UMS guest provides a front and back end service mechanism which accomplishes a number of common tasks. These tasks fall into three categories:

1. Conversion of data from external format to internal format.
2. Editing of common characteristics and enforcement of high level business rules upon input data.
3. Output conversion/formatting of internal data.

These services are grouped within a functionality unit known generically as LXTABLE. The name itself is a relation to another service which was present in a different system, and has no particular mnemonic relationship to the current system.

The intent of these services is to provide a table driven mechanism which performs common functionality, and to insure that this exists in only one place in the system. Further, the design is such that the component of the “rules” believed to be fluid is resident on the HOST computer, and downloaded to each GUEST computer at startup time each day. This allows the RMV flexibility within the rules while being able to avoid frequent GUEST installations/maintenance.

Clearly, the business rules are of primary importance to the RMV, as is the following of these rules by each GUEST. At the start of this document, the three tasks performed by LXTABLE were outlined. The first two tasks would serve this purpose in any GUEST site which is not running the native RMV GUEST. Since these services CAN be accessed within a CICS region by programs which are NOT part of the native RMV GUEST, the purpose of this package is to show how this can be done with an example program written in COBOL.

In a native UMS GUEST application, the LXTABLE services are invoked automatically by the dispatch/control services. The application program is not invoked if errors are detected. In a similar fashion, the conversion for output is automatic and transparent to the application. If the reader tries to compare the sample COBOL program with any UMS application, the conclusion is that there is no similarity. The reason for this is that in a NON UMS application, the application program is required to perform various functions which normally are the responsibility of the dispatch/control function.
In essence, the functionality of LXTABLE is simple. The programmer must construct an editing/conversion template (known as the LXTABLE). This load module is truly a table which relates input areas (raw data) with storage locations for processing (in a COMMAREA), describes how they are to be converted as well as describes what generic class of editing is to be applied.

When the editing function is invoked, the relations are processed based on the table. Conversions are performed, default values may be inserted and editing specifications are checked. If errors are detected, each is flagged along with the error-code applicable to the first detected is returned. First detected relates to the order in which the fields were specified in the LXTABLE. Normally, in the RMV usage, this relates to left-to-right, top-to-bottom construct of the input screen. This is a convention, not a technical requirement.

The process can handle all fields at once (as in the RMV usage), or process singular groups of fields. The minimum number of fields in a group is those needed to analyze all involved editing specifications. In the sample, which has two cases, one case (policy dates) requires two fields and the other (registration) requires three fields.

If many groups of fields are processed together and it is important not merely to know if ALL fields passed the edits (as in the sample), but which DID NOT, there is a simple technique. Note that in the sample, an item called EDIT-THIS-FIELD is moved to a 3 byte KEY field for each item to be processed. If the status returns non-zero, each KEY field which corresponds to an item which failed the edit will no longer have a value equal to EDIT-THIS-FIELD.
Sample Map, Non-UMS LXTABLE Invocation

= BMS map used by sample program. Provided for reference only. =
= it has NO applicability to the use of the technique. =

USAMPLE  DFHMSD
   TYPE=MAP,
   CTRL=(FREEKB,FRSET),
   MODE=INOUT,
   STORAGE=AUTO,
   TERM=3270-2,
   TIOAPFX=YES

USAMPLE  DFHMID
   SIZE=(24,80)
   DFHMDF POS=(001,023),LENGTH=0036,ATTRB=(ASKIP,BRT),
   INITIAL='EXAMPLE OF USING UMS EDIT
   X00140000
   DFHMDF POS=(002,027),LENGTH=0028,ATTRB=(ASKIP,BRT),
   INITIAL='FROM A NON-UMS APPLICATION'
   *
   DFHMDF POS=(005,001),LENGTH=13,INITIAL='POLICY EFFDT:',
   ATTRB=ASKIP
   FLD005A  DFHMDF POS=(005,015),LENGTH=10,ATTRB=(UNPROT,IC,FSET)
   DFHMDF POS=(005,026),LENGTH=07,INITIAL=' EXPDT:',
   ATTRB=ASKIP
   FLD005B  DFHMDF POS=(005,034),LENGTH=10,ATTRB=(UNPROT,FSET)
   DFHMDF POS=(005,045),LENGTH=1,INITIAL=' ',ATTRB=ASKIP
   *
   DFHMDF POS=(007,001),LENGTH=09,INITIAL='REG TYPE:',
   ATTRB=ASKIP
   FLD007A  DFHMDF POS=(007,011),LENGTH=3,ATTRB=(UNPROT,FSET)
   DFHMDF POS=(007,015),LENGTH=09,INITIAL=' NUMBER:',
   ATTRB=ASKIP
   FLD007B  DFHMDF POS=(007,025),LENGTH=7,ATTRB=(UNPROT,FSET)
   DFHMDF POS=(007,033),LENGTH=08,INITIAL=' COLOR:',
   ATTRB=ASKIP
   FLD007C  DFHMDF POS=(007,042),LENGTH=1,ATTRB=(UNPROT,FSET)
   DFHMDF POS=(007,044),LENGTH=1,INITIAL=' ',ATTRB=ASKIP
   *
   FLD014A  DFHMDF POS=(014,001),LENGTH=79,ATTRB=ASKIP,INITIAL=' '
   FLD015A  DFHMDF POS=(015,001),LENGTH=79,ATTRB=ASKIP,INITIAL=' '
   DFHMSD TYPE=FINAL
END
Program Sample, Non-UMS LXTABLE Invocation

IDENTIFICATION DIVISION.
PROGRAM-ID. USAMPLEP.
DATE-COMPILED.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
  01 DATA-WORK-ITEMS.
    04 TEMP PIC S9(9) COMP.
    04 MAPAREA.

00190005
  08 MA-TIOT-PREFIX PIC X(12).
  08 MA-FLD005A.
    12 MA-FLD005A-LEN PIC S9(4) COMP.
    12 MA-FLD005A-ATTR PIC X.
    12 MA-FLD005A-DATA PIC X(10).
  08 MA-FLD005B.
    12 MA-FLD005B-LEN PIC S9(4) COMP.
    12 MA-FLD005B-ATTR PIC X.
    12 MA-FLD005B-DATA PIC X(10).
  08 MA-FLD007A.
    12 MA-FLD007A-LEN PIC S9(4) COMP.
    12 MA-FLD007A-ATTR PIC X.
    12 MA-FLD007A-DATA PIC X(3).
  08 MA-FLD007B.
    12 MA-FLD007B-LEN PIC S9(4) COMP.
    12 MA-FLD007B-ATTR PIC X.
    12 MA-FLD007B-DATA PIC X(7).
  08 MA-FLD007C.
    12 MA-FLD007C-LEN PIC S9(4) COMP.
    12 MA-FLD007C-ATTR PIC X.
    12 MA-FLD007C-DATA PIC X(1).
  08 MA-FLD014A.
    12 MA-FLD014A-LEN PIC S9(4) COMP.
    12 MA-FLD014A-ATTR PIC X.
    12 MA-FLD014A-DATA PIC X(79).
  08 MA-FLD015A.
    12 MA-FLD015A-LEN PIC S9(4) COMP.
    12 MA-FLD015A-ATTR PIC X.

these items are relative to this sample program only, and not the general technique.

Appendix A – Non-UMS LXTABLE Edit Example
204
12 MA-FLD015A-DATA PIC X(79).
04 END-MESSAGE PIC X(27) VALUE 'END OF EDIT-SAMPLE SESSION'.
04 MAPLENGTH PIC S9(4) COMP VALUE +222.
04 CONSTANT-1 PIC S9(4) COMP VALUE +125.
04 FILLER REDEFINES CONSTANT-1.

00530001
08 FILLER PIC X.
08 HEX-7D PIC X.
^

these items are constants applicable to the RMV data structures

V
04 CONSTANT-2 PIC S9(9) COMP VALUE +32768.
04 FILLER REDEFINES CONSTANT-2.

00570005
08 FILLER PIC XX.
08 NULL-DATE PIC XX.
04 CONSTANT-3 PIC S9(9) COMP VALUE +256.
04 FILLER REDEFINES CONSTANT-3.
08 FILLER PIC X.

00620008
08 EDIT-THIS-FIELD PIC XXX.

this is the structure which defines the area containing the raw data to edit and convert. the 93 byte area at the start is required by the technique. the other items must line up with the template (LXTABLE). the 3byte field (....-key) preceeding each data item is required

V
04 DUMMY-AREA.
08 DA-HEADER PIC X(93).

00650007
08 DA-FLD005A.
12 DA-FLD005A-KEY PIC X(3).
12 DA-FLD005A-DATA PIC X(10). 00680006
 08 DA-FLD005B. 00690006
12 DA-FLD005B-KEY PIC X(3). 00700006
12 DA-FLD005B-DATA PIC X(10). 00710006
 08 DA-FLD007A. 00720006
12 DA-FLD007A-KEY PIC X(3). 00730006
12 DA-FLD007A-DATA PIC X(3). 00740006
 08 DA-FLD007B. 00750006
12 DA-FLD007B-KEY PIC X(3). 00760010
12 DA-FLD007B-DATA PIC X(7). 00770006
 08 DA-FLD007C. 00780006
12 DA-FLD007C-KEY PIC X(3). 00790010
12 DA-FLD007C-DATA PIC X(1). 00800006

------------------------------
these items are relative to this sample program only,
and not the general technique
------------------------------

04 MSG1. 00810006
 08 FILLER PIC X(8) VALUE ' EFFDT:'. 00820006
 08 M1F1. 00830006
 12 M1F1-MONTH PIC 99. 00840009
 12 M1F1-DASH1 PIC X. 00850006
 12 M1F1-DAY PIC 99. 00860009
 12 M1F1-DASH2 PIC X. 00870006
 12 M1F1-YEAR PIC 9999. 00880006
 08 FILLER PIC X(8) VALUE ' EXPDT:'. 00890006
 08 M1F2. 00900006
 12 M1F2-MONTH PIC 99. 00910009
 12 M1F2-DASH1 PIC X. 00920006
 12 M1F2-DAY PIC 99. 00930009
 12 M1F2-DASH2 PIC X. 00940006
 12 M1F2-YEAR PIC 9999. 00950006
 08 FILLER PIC X VALUE '. '. 00960006
 08 M1F3. 00970006
 12 M1F3-ERROR PIC X(6). 00980006
 12 M1F3-CODE PIC 9(9). 00990006

04 MSG2. 01000010
 08 FILLER PIC X(8) VALUE ' REGNO:'. 01010010
 08 M2F1-PREFIX PIC X(4). 01020010
 08 M2F1-REGNO PIC X(8). 01030010
 08 M2F1-COLOR PIC X(2). 01040010
 08 FILLER PIC X(14) VALUE SPACES. 01050010
 08 FILLER PIC X VALUE '. '. 01060010
08 M2F3. 01070010
12 M2F3-ERROR PIC X(6). 01080010
12 M2F3-CODE PIC 9(9). 01090010

===============================
LINKAGE SECTION. 01100000
===============================
these items are relative to =
this sample program only, =
and not the general technique =
NOTE that a BLL cell for a UMS= 
GUEST common area is needed =
but its position depends on =
the actual program structure = 

v
01 DFHCOMMAREA. 01110000
04 FILLER PIC X(4). 01120000
01 FILLER. 01130000
05 FILLER PIC S9(9) USAGE COMP. 01140000
05 UMS-COMMAREA-ADDRESS PIC S9(9) USAGE COMP.

===============================
this copybook (or an equiv- =
alent) is required. =

01 UGCOMMON-COMMAREA. 01160001
02 UGZCOMMA. 01170001
COPY UGZCOMMC. 01180001

===============================
this filler provides access to=
a field normally reserved for =
dispatch/control and also =
positions for definitions of =
the output fields from the =
edit process. =

v
02 FILLER REDEFINES UGZCOMMA. 01190001
04 FILLER PIC X(16). 01200001
04 UGCALLON PIC X. 01210001
04 FILLER PIC X(1767).
01220001
* OUTPUT DATA FIELDS COME HERE. THEY MUST BE AT THE SAME OFFSETS
* WITHIN GUEST COMMON AS EXPRESSED IN THE ASSOCIATED LXTABLE.
* EACH IS PRECEDED BY ITS Z AND T FIELD BYTES.

these fields receive the output from the edit process. the "...Z" and "...T" fields are required by the technique.
the "Z" fields may be useful for the purposes of editing from a non-UMS application, but the "T" fields are not.

```
04 SAEFFDT-GROUP.
  01260006
    08 SAEFFDT-Z PIC X. 01270006
    08 SAEFFDT-T PIC X. 01280006
    08 SAEFFDT PIC XX. 01290006
  04 SAEXPDT-GROUP.
  01300006
    08 SAEXPDT-Z PIC X. 01310006
    08 SAEXPDT-T PIC X. 01320006
    08 SAEXPDT PIC XX. 01330006
  04 SAPREFIX-GROUP.
  01340006
    08 SAPREFIX-Z PIC X. 01350006
    08 SAPREFIX-T PIC X. 01360006
    08 SAPREFIX PIC X(3). 01370006
  04 SAREGNO-GROUP.
  01380006
    08 SAREGNO-Z PIC X. 01390006
    08 SAREGNO-T PIC X. 01400006
    08 SAREGNO PIC X(7).
  04 SACOLOR-GROUP.
  01410006
    04 SACOLOR-GROUP.
  01420006
    08 SACOLOR-Z PIC X. 01430006
    08 SACOLOR-T PIC X. 01440006
    08 SACOLOR PIC X(1).
  01450010
```

```
PROCEDURE DIVISION.
  0000-SETUP-PROGRAM.
```
this code is just relative to the sample program and has no relevance to the technique.

**** TEST FOR FIRST-TIME THROUGH,
**** SEND FIRST EMPTY SCREEN IF SO:
01480001
  IF EIBCALEN = ZERO
    EXEC CICS SEND MAP('USAMPLE') MAPONLY ERASE FREEKB END-EXEC
  EXEC CICS RETURN COMMAREA(TEMP) LENGTH(4) TRANSID(EIBTRNID) END-EXEC.

******* INPUT MAP:
01510001
  EXEC CICS RECEIVE MAP('USAMPLE') INTO(MAPAREA) NOHANDLE END-EXEC.

******* TEST TERMINATION (ANY KEY EXCEPT ENTER):
01530001
  IF EIBAID NOT = HEX-7D
01540001
    EXEC CICS SEND TEXT FROM(END-MESSAGE) CURSOR(1) LENGTH(27) ERASE FREEKB END-EXEC
01560001
  EXEC CICS RETURN END-EXEC.

-----------------------------
UIGETCOM returns a skeleton UMS common area with a number of address constants plugged in

******* GET A UMS-GUEST COMMON AREA:
01620001
  EXEC CICS LINK PROGRAM('UIGETCOM') COMMAREA(UMS-COMMAREA-ADDRESS)
01630001
  LENGTH(4) END-EXEC.

-----------------------------

tell COBOL that a related BLL cell has changed

  SERVICE RELOAD UGCOMMON-COMMAREA.
****** INITIALIZE CONSTANTS NORMALLY PROVIDED BY
DISPATCH-CONTROL
=================================================================
the edit functions require the=
current date in RMV format =
get it from CICS in Julian =
format and use the UMS service=
module to convert it to the =
RMV format =

MOVE EIBDATE TO UGCOMMON-DATE-JULIAN-FORMAT. 01680005
MOVE '2' TO UGCOMMON-DATE-INPUT-FORMAT. 01690005
CALL 'UICALLST' USING UGCOMMON-DATE-ROUTINE-ADDRESS
UGCOMMON-COMMAREA. 01710005
MOVE UGCOMMON-DATE-BINARY-FORMAT TO 01720005
UGCOMMON-CURRENT-BINARY-DATE. 01730005

=================================================================

=================================================================
setup items required to show =
that the next phase is edit =

MOVE +01 TO UGCOMMON-ENTRY-REASON. 01740005
MOVE 'Y' TO UGCOMMON-MESSAGE-TEXT. 01750005

=================================================================

=================================================================
this code is just relative =
to the sample program and has =
no relevance to the technique.=

****** DO SOME PROCESSING RELATIVE TO
******   THE TEST MAP WE JUST INPUT:
******

MOVE SPACES TO MA-FLD014A-DATA, MA-FLD015A-DATA.
IF MA-FLD005A-DATA = (LOW-VALUES OR SPACES) AND
MA-FLD005B-DATA = (LOW-VALUES OR SPACES) AND
MA-FLD007A-DATA = (LOW-VALUES OR SPACES) AND
MA-FLD007B-DATA = (LOW-VALUES OR SPACES) AND
MA-FLD007C-DATA = (LOW-VALUES OR SPACES)
MOVE 'NO DATA WAS ENTERED' TO MA-FLD014A-DATA
GO TO 9999-EXIT. 01840005

=================================================================

=================================================================

Appendix A – Non-UMS LXTABLE Edit Example

DUMMY-AREA will contain data to be edited. In fact, the edit services will treat it as if it were an input map composite. A valid length is required.

******* CONSTANT BELOW SHOULD REPRESENT THE TRUE LENGTH OF DUMMY-AREA

MOVE +58 TO UGCOMMON-RECEIVED-MAP-LENGTH.

------------------------------------------------------------------

since DUMMY-AREA is equivalent to a map, its address is required in both map address cells in the common area. UMS services can be used to get this value:

V

******* ESTABLISH ADDRESSES OF DUMMY MAP AREA:01870006

CALL 'UICALLST' USING UGCOMMON-DATA-NAME-ADDRESS

DUMMY-AREA, UGCOMMON-INPUT-MAP-ADDRESS.

MOVE UGCOMMON-INPUT-MAP-ADDRESS TO UGCOMMON-OUTPUT-MAP-ADDRESS.

------------------------------------------------------------------

the name of the user constructed edit template (LXTABLE) must be placed in the common area

V

******* SET EDIT-TABLE NAME: 01920006

MOVE 'USAMPLET' TO UGCOMMON-MAP-TRANSLATION-TBL.

------------------------------------------------------------------

this code is just relative to the sample program and has no relevance to the technique.

V

******* 01940006

******* 01950006
***** PROCESS EFFDT/EXPDT: 01960006
*****

IF MA-FLD005A-DATA = (LOW-VALUES OR SPACES) AND
MA-FLD005B-DATA = (LOW-VALUES OR SPACES)
MOVE 'NO POLICY DATE(S)' TO MA-FLD014A-DATA
GO TO 1000-DO-REGNO.

the edit services normally are used to process many fields at the same time. this technique edits all fields, but the returned code is only for the 1st error, with the checking in the order the definitions occur in the template. this example makes multiple passes to show all codes. the verbage documentation describes how to tell which fields are in error if multiple fields are edited at once.

***** THE NEXT 4 LINES OF CODE ARE IN ORDER FOR EACH EDITING PASS:

MOVE LOW-VALUES TO UGCOMMON-PROGRAM-COMMAREA.
MOVE HIGH-VALUES TO UGCALLON.
MOVE ZERO TO UGCOMMON-CURRENT-ERROR-CODE,
UGCOMMON-CURRENT-MAP-LENGTH.

the conversion output fields must be set to the default value for the field mode.

***** INITIALIZE INVOLVED FIELDS TO DEFAULT VALUE(S)

MOVE NULL-DATE TO SAEFFDT, SAEXPDT.

Appendix A – Non-UMS LXTABLE Edit Example
only the fields to edit are set in DUMMY-AREA. they must be flagged as edit candidates.

****** CLEAR DUMMY-AREA, PLUG INVOLVED FIELDS:
    MOVE LOW-VALUES TO DUMMY-AREA.
    MOVE MA-FLD005A-DATA TO DA-FLD005A-DATA.
    MOVE MA-FLD005B-DATA TO DA-FLD005B-DATA.

****** INDICATE WHICH FIELDS ARE TO BE PROCESSED:
    MOVE EDIT-THIS-FIELD TO DA-FLD005A-KEY
    DA-FLD005B-KEY.

this code is just relative to the sample program and has no relevance to the technique.

MOVE SPACES TO MA-FLD005A-DATA, MA-FLD005B-DATA.

invoke the edit process. note that a UGCA abend means that the UMS commarea was not obtained properly. a UGTL abend means that there is not a running UMS guest in the region. UGTL only occurs if the particular edit requires host download data.

****** INVOKE THE EDIT PROCESS:
    EXEC CICS LINK PROGRAM('UGZ0006P')
    COMMAREA(UGCOMMON-COMMAREA)
    LENGTH(UGCOMMON-XCTL-LENGTH) END-EXEC.

this code is just relative to the sample program and has no relevance to the technique. it is worthy of observation because it demonstrates the
UMS field cascading technique, testing the error-code field and converting binary dates for output. Note that date conversion specifies the input format and all other formats are always returned. In a binary conversion the status will always be valid.

V


******* ASSUME THAT EDIT/REFORMAT MIGHT HAVE CHANGED THE MAP IMAGE:

MOVE DA-FLD005A-DATA TO M1F1.
MOVE DA-FLD005B-DATA TO M1F2.

******* CONVERT INTERNAL FIELDS TO DISPLAY IF THEY WERE INPUT OK:

IF SAEFFDT NOT = NULL-DATE
MOVE SAEFFDT TO UGCOMMON-DATE-BINARY-FORMAT
MOVE '0' TO UGCOMMON-DATE-INPUT-FORMAT
CALL 'UICALLST' USING UGCOMMON-DATE-ROUTINE-ADDRESS UGCOMMON-COMMAREA
MOVE UGCOMMON-DATE-GREG-DAY TO M1F1-DAY
MOVE UGCOMMON-DATE-GREG-MONTH TO M1F1-MONTH
MOVE UGCOMMON-DATE-GREG-YEAR TO M1F1-YEAR
MOVE '/' TO M1F1-DASH1, M1F1-DASH2.

IF SAEXPDT NOT = NULL-DATE
MOVE SAEXPDT TO UGCOMMON-DATE-BINARY-FORMAT
MOVE '0' TO UGCOMMON-DATE-INPUT-FORMAT
CALL 'UICALLST' USING UGCOMMON-DATE-ROUTINE-ADDRESS UGCOMMON-COMMAREA
MOVE UGCOMMON-DATE-GREG-DAY TO M1F2-DAY
MOVE UGCOMMON-DATE-GREG-MONTH TO M1F2-MONTH
MOVE UGCOMMON-DATE-GREG-YEAR TO M1F2-YEAR
MOVE '/' TO M1F2-DASH1, M1F2-DASH2.

******* DISPLAY "OK" OR THE ERROR-CODE:

IF UGCOMMON-CURRENT-ERROR-CODE = ZERO
MOVE 'OK' TO M1F3
ELSE
MOVE 'ERROR ' TO M1F3-ERROR
MOVE UGCOMMON-CURRENT-ERROR-CODE TO M1F3-CODE.

******* PUT THE MESSAGE IN THE OUTPUT MAP
MOVE MSG1 TO MA-FLD014A-DATA.

1000-DO-REGNO.

*******

******* PROCESS REG PREFIX, NUMBER, COLOR:

********

this code is just relative to the sample program and has no relevance to the technique.

\[\text{v}\]

IF MA-FLD007A-DATA = (LOW-VALUES OR SPACES) AND
   MA-FLD007B-DATA = (LOW-VALUES OR SPACES) AND
   MA-FLD007C-DATA = (LOW-VALUES OR SPACES)
   MOVE 'NO REGISTRATION DATA' TO MA-FLD015A-DATA
   GO TO 9999-EXIT.

\[\text{v}\]

the edit services normally are used to process many fields at the same time. this technique edits all fields, but the returned code is only for the 1st error, with the checking in the order the definitions occur in the template. this example makes multiple passes to show all codes. the verbage documentation describes how to tell which fields are in error if multiple fields are edited at once.

\[\text{v}\]

******* THE NEXT 4 LINES OF CODE ARE IN ORDER FOR EACH EDITING PASS:

\[\text{v}\]

MOVE LOW-VALUES TO UGCOMMON-PROGRAM-COMMAREA.
MOVE HIGH-VALUES TO UGCALLON.
MOVE ZERO TO UGCOMMON-CURRENT-ERROR-CODE,
   UGCOMMON-CURRENT-MAP-LENGTH.
the conversion output fields = must be set to the default = value for the field mode. =

******** INITIALIZE INVOLVED FIELDS TO DEFAULT VALUE(S)
MOVE SPACES TO SAPREFIX, SAREGNO, SACOLOR.

===============================================

only the fields to edit are = set in DUMMY-AREA. they must = be flagged as edit candidates. =

******** CLEAR DUMMY-AREA, PLUG INVOLVED FIELDS:
MOVE LOW-VALUES TO DUMMY-AREA.
MOVE MA-FLD007A-DATA TO DA-FLD007A-DATA.
MOVE MA-FLD007B-DATA TO DA-FLD007B-DATA.
MOVE MA-FLD007C-DATA TO DA-FLD007C-DATA.

******** INDICATE WHICH FIELDS ARE TO BE PROCESSED:
MOVE EDIT-THIS-FIELD TO DA-FLD007A-KEY
DA-FLD007B-KEY
DA-FLD007C-KEY.

===============================================

this code is just relative = to the sample program and has = no relevance to the technique. =

MOVE SPACES TO MA-FLD007A-DATA, MA-FLD007B-DATA
MA-FLD007C-DATA.

===============================================

invoke the edit process. note = that a UGCA abend means that = the UMS commarea was not = obtained properly. a UGTL = abend means that there is not = a running UMS guest in the = region. UGTL only occurs if = the particular edit requires = host download data. =
******* INVOKE THE EDIT PROCESS:
  EXEC CICS LINK    PROGRAM('UGZ0006P')
  COMMAREA(UGCOMMON-COMMAREA)
  LENGTH(UGCOMMON-XCTL-LENGTH) END-EXEC.
^ 
=================================================================
this code is just relative    =
to the sample program and has =
no relevance to the technique.=
it is worthy of observation .=
because it demonstrates the    =
UMS field cascading technique,=
and testing the error-code    =
field.        =

  

=================================================================

******* THE REST OF THIS PARAGRAPH SIMPLY FORMATS FOR THE SAMPLE
******* DISPLAY. THE ESSENCE OF THE LOGIC IS THAT
******* IF UGCOMMON-CURRENT-ERROR-CODE = ZERO,
******* THE EDIT WAS PASSED
******* AND THE CONVERTED CORE-IMAGE WAS BUILT.
*******
******* ASSUME THAT EDIT/REFORMAT MIGHT HAVE CHANGED
******* THE MAP IMAGE:
    MOVE DA-FLD007A-DATA TO M2F1-PREFIX.
    MOVE DA-FLD007B-DATA TO M2F1-REGNO.
    MOVE DA-FLD007C-DATA TO M2F1-COLOR.
******* CONVERT INTERNAL FIELDS TO DISPLAY IF THEY WERE INPUT
OK:
    IF SAPREFIX NOT = SPACES MOVE SAPREFIX   TO M2F1-PREFIX.
    IF SAREGNO NOT = SPACES MOVE SAREGNO    TO M2F1-REGNO.
    IF SACOLOR NOT = SPACES MOVE SACOLOR    TO M2F1-COLOR.
******* DISPLAY "OK" OR THE ERROR-CODE:
    IF UGCOMMON-CURRENT-ERROR-CODE = ZERO
      MOVE 'OK' TO M2F3
    ELSE
      MOVE 'ERROR ' TO M2F3-ERROR
      MOVE UGCOMMON-CURRENT-ERROR-CODE TO M2F3-CODE.
******* PUT THE MESSAGE IN THE OUTPUT MAP
    MOVE MSG2 TO MA-FLD015A-DATA.
^ 
=================================================================
9999-EXIT.
=================================================================
cleanup of the common area    =
is important, in particular if=
the logic is going in some resident program which hangs on data availability. the UIGETCOM routine will return new areas on successive calls.

EXEC CICS FREEMAIN DATA(UGCOMMON-COMMAREA) END-EXEC.

this code is just relative to the sample program and has no relevance to the technique.

MOVE 'A' TO MA-FLD005A-ATTR, MA-FLD005B-ATTR, MA-FLD007A-ATTR, MA-FLD007B-ATTR, MA-FLD007C-ATTR.

EXEC CICS SEND MAP('USAMPLE') DATAONLY NOHANDLE FROM(MAPAREA) LENGTH(MAPLENGTH) FREEKB END-EXEC.

EXEC CICS RETURN COMMAREA(TEMP) LENGTH(4) TRANSID(EIBTRNID) END-EXEC.

GOBACK.
Sample LXTABLE, Non-UMS Invocation

= editing TEMPLATE (LXTABLE) used by sample program =

=============================================
required:  =
  ^
  TITLE 'USAMPLET - UMS LX SCREEN ' 00092000
  COPY UMSLXMAC 00093000
  PUSH PRINT 00094000
  PRINT OFF 00095000
  COPY UGZCOMMA 00096000
  POP PRINT 00097000
  ORG UGCUSERA 00098100
  ~

=============================================

conversion output fields. =
this definition must match =
the definition (format, =
length and offset) in the =
invoking program. =
(UMS commarea) =

  ^
  SAEFFDT GFLD HL2 EFFECTIVE-DATE 00098300
  SAEXPDT GFLD HL2 EXPIRATION-DATE 00098400
  SAPREFIX GFLD CL3 PREFIX 00098500
  SAREGNO GFLD CL7 REGNO 00098600
  SACOLOR GFLD CL1 COLOR 00098700
  ORG 00104400
  * 00104500
  ~

=============================================

required, name in col-1 =
must be load-module name. =
NOMAPNA should be used for =
all non-UMS standard usage =

  ^
  EJECT 00104600
USAMPLET UMSLXTBL TYPE=START,MAPNAME=NOMAPNA,LEVEL0=NO 00105000
^  
=-------------------------------------------------------------=

  date specifications. the =
  effective-date happens to =
  be first, and the expire =
  date second. the editing =
  specification MUST be on =
  the last entry. =

  =

  UMSLXTBL MAPFLD=FLD005A,GSAFLD=SAEFFDT  00110000
  UMSLXTBL MAPFLD=FLD005B,GSAFLD=SAEXPDT,
  EDIT=(PDATE,SAEFFDT,SAEXPDT)  00130000
  ^

=-------------------------------------------------------------=

  reg specifications. the =
  order happens to be prefix,=
  regno and color. the =
  editing specification MUST =
  be on the last entry. =

  =

  UMSLXTBL MAPFLD=FLD007A,GSAFLD=SAPREFIX  00140000
  UMSLXTBL MAPFLD=FLD007B,GSAFLD=SAREGNO  00141000
  UMSLXTBL MAPFLD=FLD007C,GSAFLD=SACOLOR,
  X00142000
  EDIT=(REGNO,SAPREFIX,SACOLOR,SAREGNO)  00160000
  EJECT  00490000
  ^

=-------------------------------------------------------------=

  required: =

  =

  UMSHEADR TYPE=DSECT  00500000
  ^

=-------------------------------------------------------------=

  conversion input fields. =
  this definition must match =
  the definition (format, =
  length and offset) in the =
  invoking program. =
  (DUMMY-AREA) =
DS CL3      BMS DATA 00510000
FLD005A DS CL10        EFFDT 00520000
  DS CL3      BMS DATA 00521000
FLD005B DS CL10        EXPDT 00522000
  DS CL3      BMS DATA 00530000
FLD007A DS CL3         PREFIX 00540000
  DS CL3      BMS DATA 00550000
FLD007B DS CL7         REGNO 00560000
  DS CL3      BMS DATA 00570000
FLD007C DS CL1         COLOR 00580000

====================================================================

required: =

END  01260300

====================================================================
Appendix B:

Guest Process Control Table

The macro prototype is included here for ease of understanding. Refer to Chapter 5 for a
discussion of the parameters. The current PTEG table is provided for illustration and
system structure documentation.

```
PTEG &NAME,&PGM=,&TASK=,&XLATE=,&FLAGS=,&CLEAN=,
    &ALIAS=,&SCROLL=NO,&RESCROLL=NO,&UPDATE=NO,
    &HOSTA=NO,&XFER=NO,&PF4=,&PF9=,&PF7AND8=NO,
    &DUPKEY=,&INTERNL=NO
```

The table header and footer are shown for completeness.

```
UGZ0002P CSECT                                                            05900000
NUMENT   DC    F'0'            NUMBER OF ENTRIES                        05910000
       DC    A(FIRST)        ADDRESS OF FIRST ENTRY                   05920000
NUMXFER  DC    F'0'            NUMBER OF XFER (DUPKEY) ENTRIES          05930000
       DC    A(XFER1)        ADDRESS OF FIRST XFER ENTRY              05940000
NUMALIAS DC    F'0'            NUMBER OF ALIAS ENTRIES                  05950000
       DC    A(ALIAS1)       ADDRESS OF FIRST ALIAS ENTRY             05960000
UGZ0002Q CSECT                                                            05970000
       XFER1    EQU   *               FIRST XFER ENTRY                         05980000
UGZ0002R CSECT                                                            05990000
       ALIAS1   EQU   *               FIRST ALIAS ENTRY                        06000000
UGZ0002P CSECT                                                            06010000
       *                                                                         06020000
       FIRST    DS    0F
.AA020   ANOP
* WARNING: NEXT ENTRY HAS UNPRINTABLES IN THE FUNCTION-CODE                  06030000
* IT IS USED FOR INTERNAL INVOKATION FROM ALAR                                 06040000
PCTEG 'A ',PGM=UGZ0012P,TYPE=RMV                                             06050000
PCTEG 'ADM',PGM=UGZ0009P,TASK=ADM,RESCROLL=NO,TASK=RMV                      06060000
* AUTO EXAM BOOKING 10/12/92                                               06070000
PCTEG 'AEB ',PGM=(UGB0213P),TYPE=RMV,INTERNL=YES,UPDATE=YES,HOSTA=YES        06080000
PCTEG 'ALAR ',PGM=(UGB0213P),TYPE=RMV,INTERNL=YES,UPDATE=YES,HOSTA=YES       06090000
* EXAM BOOKING - DRIVING SCHOOL BLOCK BOOKING                                06100000
PCTEG 'BKBB ',PGM=(UGB0500P,UGB0501P),XFER=UGB0500T,UPDATE=YES,SCROLL=YES,HOSTA=YES 06110000
```

Appendix B – Guest Process Control Table
* EXAM BOOKING - SITE CLOSURE EXTERNAL AND INTERNAL FUNCTIONS  
PCTEG 'BKCL', PGM=(UGB0410P, UGB0411P), XLATE=UGB0410T,  
   UPDATE=YES, SCROLL=YES, HOSTA=YES 06160000  
   X06170000  
PCTEG 'BK1', PGM=(UGB0420P, UGB0421P), XLATE=UGB0420T,  
   UPDATE=YES, SCROLL=YES, HOSTA=YES 06190000  
   X06200000  
PCTEG 'BK2', PGM=(UGB0430P, UGB0431P), XLATE=UGB0430T,  
   UPDATE=YES, SCROLL=YES, HOSTA=YES 06210000  
   X06220000  
* EXAM BOOKING - REDESIGN - EXTERNAL AND INTERNAL FUNCTIONS  
PCTEG 'BK01', PGM=(UGB0310P, UGB0311P), XLATE=UGB0310T,  
   SCROLL=YES, HOSTA=YES, PF4=4, PF9=4 06240000  
   X06250000  
PCTEG 'BK02', PGM=(UGB0320P, UGB0321P), XLATE=UGB0320T,  
   SCROLL=YES, HOSTA=YES, INTERNL=YES 06260000  
   X06270000  
PCTEG 'BK03', PGM=(UGB0330P, UGB0331P), XLATE=UGB0330T,  
   UPDATE=YES, HOSTA=YES, INTERNL=YES 06280000  
   X06290000  
PCTEG 'BK04', PGM=(UGB0340P, UGB0341P), XLATE=UGB0340T,  
   UPDATE=YES, HOSTA=YES, INTERNL=YES 06300000  
   X06310000  
* CCP - CMVI CITATION (INQ BY CITATION)  
PCTEG 'CCP ', PGM=(UGC0011P, UGC0012P), XLATE=UGC0010T,  
   SCROLL=NO, RESCROLL=NO, UPDATE=YES, HOSTA=YES, PF4=A045,  
   PF9=A045, PF7AND8=YES, NULL9=YES 06320000  
   X06340000  
* CCR - CREDIT CARD RECONCILIATION  
PCTEG 'CCR ', PGM=(UGC0611P, UGC0612P), XLATE=UGC0610T,  
   UPDATE=YES, HOSTA=YES, NULL9=NO, PF4=012, PF9=012,  
   SCROLL=NO 06360000  
   X06380000  
* CIC - PAYMENT HISTORY SCROLL  
PCTEG 'CIC ', PGM=(UGC1041P, UGC1042P), XLATE=UGC1040T,  
   SCROLL=YES, RESCROLL=YES, HOSTA=YES, PF4=A045,  
   PF9=A 06400000  
   X06420000  
PCTEG 'CJRI', PGM=(UGR1061P, UGR1062P), XLATE=UGR1060T,  
   HOSTA=YES, PF4=1024, PF9=1024 06440000  
   X06460000  
* COR - OBLIGATION RESOLUTION SUMMARY SCREEN  
PCTEG 'COR ', PGM=(UGS1191P, UGS1192P), XLATE=UGS1190T,  
   UPDATE=YES, HOSTA=YES, NULL9=YES, PF4=04, PF9=045,  
   SCROLL=YES 06490000  
   X06510000  
* COR1 - OBLIGATION RESOLUTION DETAIL SCREEN  
PCTEG 'COR1', PGM=(UGS1201P, UGS1202P), XLATE=UGS1200T,  
   HOSTA=YES, NULL9=YES, PF9=045, SCROLL=YES,  
   INTERNL=YES 06520000  
   X06540000  
* CPB - BACKOUT TRANSACTIONS  
PCTEG 'CPB ', PGM=(UGC1061P, UGC1062P), XLATE=UGC1060T,  
   UPDATE=YES, PF4=1, HOSTA=YES, TYPE=RMV 06550000  
   X06570000  
* CPD - RETURNED CHECK RESOLUTION  
PCTEG 'CPD ', PGM=(UGC1031P, UGC1032P), XLATE=UGC1030T,  
   SCROLL=NO, RESCROLL=NO, UPDATE=YES, HOSTA=YES, PF4=V 06590000  
   X06610000  
* CPG - NEW CASH SCREEN FOR FAST LANE  
PCTEG 'CPG ', PGM=(UGC1051P, UGC1052P), XLATE=UGC1050T,  
   UPDATE=YES, PF4=0, PF9=0, HOSTA=YES, PF7AND8=YES, TYPE=RMV 06630000  
   X06640000  
* DRAD - ADDRESS ADD AND AMEND FOR SECTION V  
PCTEG 'DRAD', PGM=(UGR1810P, UGR1812P), XLATE=UGR1810T,  
   HOSTA=YES, TYPE=RMV 06650000  
   X06670000  
* DRPL - PLATE ADD AND AMEND FOR SECTION V  
PCTEG 'DRPL', PGM=(UGR1800P, UGR1802P), XLATE=UGR1800T,  
   HOSTA=YES, TYPE=RMV 06690000  
   X06700000  
* EXAM BOOKING 12/13/91 VERSION (SOME CDL CHANGES)  
PCTEG 'EB ', PGM=(UGB0210P, UGB0211P, UGB0212P), XLATE=UGB0210T,  
   X06720000
SCROLL=YES, UPDATE=YES, HOSTA=YES, PF4=4, PF9=4

* EXAM BOOKING DISTANCE CHECKOUT:
  PCTEG 'EBDC', PGM=(UGB3040P, UGB3041P), XLATE=UGB3040T,
  HOSTA=YES

* EXAM BOOKING LOCATION MAINTENANCE:
  PCTEG 'EBLM', PGM=(UGB3020P, UGB3021P), XLATE=UGB3020T,
  HOSTA=YES, UPDATE=YES

* EXAM BOOKING OFFICE MAINTENANCE:
  PCTEG 'EBOM', PGM=(UGB3060P, UGB3061P), XLATE=UGB3060T,
  HOSTA=YES, UPDATE=YES

* EXAM BOOKING ZIP MAINTENANCE:
  PCTEG 'EBZM', PGM=(UGB3000P, UGB3001P), XLATE=UGB3000T,
  HOSTA=YES, UPDATE=YES

* EGCO - CORRECT GRADED ROAD EXAMS
  PCTEG 'EGCO', PGM=(UGB0730P, UGB0731P), XLATE=UGB0730T,
  UPDATE=YES, HOSTA=YES, PF4=4, PF9=4

* EGRD - GRADE ROAD EXAMS - INDIV GRADE
  PCTEG 'EGRD', PGM=(UGB0710P, UGB0711P), XLATE=UGB0710T,
  SCROLL=YES, UPDATE=YES, HOSTA=YES

* ESCH - EXAMINER SCHEDULE
  PCTEG 'ESCH', PGM=(UGB0700P, UGB0701P, UGB0702P), XLATE=UGB0700T,
  SCROLL=YES, UPDATE=YES, HOSTA=YES

* EVRR - EVR RESEND RESPONSE SCREEN
  PCTEG 'EVRR', PGM=(UGR1171P, UGR1172P), XLATE=UGR1170T,
  SCROLL=NO, RESCROLL=NO, UPDATE=YES, HOSTA=YES, TYPE=RMV

* FOREIGN CASH POSTING
  PCTEG 'FCP', PGM=(UGC1021P, UGC1022P), XLATE=UGC1020T,
  SCROLL=NO, UPDATE=NO, HOSTA=YES, PF7AND8=NO,
  INTERN=NO, NULL9=NO, PF5=NO

* IMI - INSPECTION HISTORY INQUIRY
  PCTEG 'IMI', PGM=(UGI1341P, UGI1342P), XLATE=UGI1340T,
  UPDATE=YES, HOSTA=YES, PF4=01, PF9=01

* IMQ - INSPECTION HISTORY UPDATE
  PCTEG 'IMQ', PGM=(UGI1321P, UGI1322P), XLATE=UGI1320T,
  UPDATE=NO, HOSTA=YES, NULL9=NO, PF4=012, PF9=012

* IMU - INSPECTION HISTORY UPDATE
  PCTEG 'IMU', PGM=(UGI1351P, UGI1352P), XLATE=UGI1350T,
  UPDATE=YES, HOSTA=YES, PF4=01, PF9=01

<table>
<thead>
<tr>
<th>Process</th>
<th>Description</th>
<th>Code</th>
<th>Parameters</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LACH</td>
<td>License Address Change History</td>
<td>PCTEG 'LACH', X = UGL3320P, UGL3321P, Xlate = UGL3320T, PF4 = 4, PF9 = 4, HOSTA = YES, PF7AND8 = YES</td>
<td>X07300000</td>
<td>07300000</td>
</tr>
<tr>
<td>LAI</td>
<td>License Address Information</td>
<td>PCTEG 'LAI ', X = UGL3300P, UGL3301P, Xlate = UGL3300T, PF4 = 4, PF9 = 4, HOSTA = YES</td>
<td>X07340000</td>
<td>07340000</td>
</tr>
<tr>
<td>LICF</td>
<td>License Fees Inquiry</td>
<td>PCTEG 'LICF', X = UGL0400P, UGL0401P, Xlate = UGL0400T, SCROLL = YES, HOSTA = YES, PF4 = 4, PF9 = 4, UPDATE = YES</td>
<td>X07490000</td>
<td>07500000</td>
</tr>
<tr>
<td>LIFI</td>
<td>License Fees Inquiry</td>
<td>PCTEG 'LIFI', X = UGL0410P, UGL0411P, Xlate = UGL0410T, SCROLL = YES, HOSTA = YES, PF4 = 4, PF9 = 4</td>
<td>X07520000</td>
<td>07530000</td>
</tr>
<tr>
<td>LIMC</td>
<td>License Issue Motorcycle Course</td>
<td>PCTEG 'LIMC', X = UGL0700P, UGL0701P, Xlate = UGL0700T, UPDATE = YES, SCROLL = NO, HOSTA = YES</td>
<td>X07550000</td>
<td>07560000</td>
</tr>
<tr>
<td>LIQI</td>
<td>Easy/Fast Way to Issue License</td>
<td>PCTEG 'LIQI', X = UGL0050P, UGL0051P, Xlate = UGL0050T, PF4 = 4, PF9 = 4, PF7AND8 = NO, TYPE = RMV, UPDATE = YES, Xlate = UGL0050T</td>
<td>X07610000</td>
<td>07620000</td>
</tr>
<tr>
<td>LIT1</td>
<td>Test Driver</td>
<td>PCTEG 'LIT1', X = UGL0206P, UGL0207P, Xlate = UGL0206T, UPDATE = YES, HOSTA = YES</td>
<td>X07680000</td>
<td>07690000</td>
</tr>
<tr>
<td>LIT2</td>
<td>Test Driver</td>
<td>PCTEG 'LIT2', X = UGL0208P, UGL0209P, Xlate = UGL0208T, UPDATE = YES, HOSTA = YES</td>
<td>X07700000</td>
<td>07710000</td>
</tr>
<tr>
<td>LIT3</td>
<td>Test Driver</td>
<td>PCTEG 'LIT3', X = UGL0210P, UGL0211P, Xlate = UGL0210T, UPDATE = YES, HOSTA = YES</td>
<td>X07720000</td>
<td>07730000</td>
</tr>
<tr>
<td>LIT4</td>
<td>Test Driver</td>
<td>PCTEG 'LIT4', X = UGL0223P, UGL0224P, Xlate = UGL0223T, UPDATE = YES, HOSTA = YES</td>
<td>X07740000</td>
<td>07750000</td>
</tr>
<tr>
<td>LIT5</td>
<td>Test Driver</td>
<td>PCTEG 'LIT5', X = UGL0218P, UGL0219P, Xlate = UGL0218T, UPDATE = YES, HOSTA = YES</td>
<td>X07760000</td>
<td>07770000</td>
</tr>
<tr>
<td>LIW</td>
<td>Warrants, Inquire for License</td>
<td>PCTEG 'LIW ', X = UGN0360P, UGN0361P, Xlate = UGN0360T, HOSTA = YES, PF4 = 4, PF9 = 4, PF7AND8 = YES</td>
<td>X07790000</td>
<td>07800000</td>
</tr>
<tr>
<td>LMFC</td>
<td>Cancel License Manufacture</td>
<td>PCTEG 'LMFC', X = UGL0480P, UGL0481P, Xlate = UGL0480T, PF4 = 4, PF9 = 4, HOSTA = YES, TYPE = RMV, UPDATE = YES</td>
<td>X07820000</td>
<td>07830000</td>
</tr>
<tr>
<td>LMH</td>
<td>License Manufacture History</td>
<td>PCTEG 'LMH ', X = UGL3200P, UGL3201P, Xlate = UGL3200T, PF4 = 4, PF9 = 4, HOSTA = YES, TYPE = RMV</td>
<td>X07850000</td>
<td>07860000</td>
</tr>
</tbody>
</table>
* LICENSE NUMBER SCROLL
PCTEG 'LN ', PGM=(UGL0020P, UGL0021P), HOSTA=YES, SCROLL=YES, X07880000
RESERVED=YES, DUPKEY=LI04, ALIAS='ULN ' 07890000

* OUT OF STATE LICENSE NUMBER SCROLL
PCTEG 'LNO ', PGM=(UGL0240P, UGL0241P), HOSTA=YES, SCROLL=YES, 07910000
RESERVED=YES, DUPKEY=LD08, XLATE=UGL0240T, ALIAS='ULNO' 07920000

* SOCIAL SECURITY NUMBER SCROLL
PCTEG 'LNS ', PGM=(UGL0220P, UGL0221P), HOSTA=YES, SCROLL=YES, 07940000
RESERVED=YES, DUPKEY=UD04, XLATE=UGL0220T, ALIAS='ULNS' 07950000

* IMAGING OOS DATA COLLECTION
PCTEG 'LOS ', PGM=(UGL0600P, UGL0601P), XLATE=UGL0600T, X07970000
UPDATE=YES, PF4=4, HOSTA=YES 07980000

* PDPS EMPLOYER/EMPLOYEE INQUIRY
PCTEG 'LPDE', PGM=(UGL0360P, UGL0361P), XLATE=UGL0360T, X08000000
UPDATE=YES, PF4=4, PF9=4, HOSTA=YES, TYPE=RMV 08010000

* PDPS MESSAGE LOG
PCTEG 'LPDH', PGM=(UGL0310P, UGL0311P), XLATE=UGL0310T, X08030000
UPDATE=YES, PF4=4, PF9=4, HOSTA=YES, TYPE=RMV 08040000

* PDPS ISSUE REQUEST
PCTEG 'LPDI', PGM=(UGL0320P, UGL0321P), XLATE=UGL0320T, X08060000
UPDATE=YES, PF4=4, PF9=4, HOSTA=YES, TYPE=RMV 08070000

* PDPS MESSAGE DETAIL - PART 1
PCTEG 'LPDR', PGM=(UGL0340P, UGL0341P), XLATE=UGL0340T, X08090000
UPDATE=YES, PF4=4, PF9=4, PF7AND8=YES, HOSTA=YES, TYPE=RMV 08100000

* PDPS MESSAGE DETAIL - PART 2
PCTEG 'LPDS', PGM=(UGL0350P, UGL0351P), XLATE=UGL0350T, X08120000
UPDATE=YES, PF4=4, PF9=4, PF7AND8=YES, HOSTA=YES, TYPE=RMV 08130000

* JOL PASSENGER RESTRICTION MAINT
PCTEG 'LPRM', PGM=(UGL3206P, UGL3207P), XLATE=UGL3206T, X08150000
UPDATE=YES, PF4=4, PF9=4, PF7AND8=NO, HOSTA=YES, TYPE=RMV 08160000

* SSN DELETE FUNCTION
PCTEG 'LSSD', PGM=(UGL1200P, UGL1201P), XLATE=UGL1200T, X08180000
HOSTA=YES, PF4=4, PF9=4, UPDATE=YES 08190000

* SSN VERIFICATION AND RESOLUTION
PCTEG 'LSSN', PGM=(UGL0870P, UGL0871P), XLATE=UGL0870T, X08210000
UPDATE=YES, HOSTA=YES, NULL9=YES, PF9=4, PF4=4, TYPE=RMV 08220000

* LICENSE TRANSACTION HISTORY
PCTEG 'LTH ', PGM=(UGL0260P, UGL0261P, UGL0262P), XLATE=UGL0260T, X08240000
PF4=4, PF9=4, HOSTA=YES, PF7AND8=YES, ALIAS='ULTH' 08250000

* LICENSE IMAGE HISTORY
PCTEG 'LTHI', PGM=(UGL0650P, UGL0651P), XLATE=UGL0650T, X08270000
PF4=4, PF9=4, HOSTA=YES, PF7AND8=YES 08280000

* LICENSE APPLY WARRANT EXEMPTION
PCTEG 'LWA', PGM=(UGN0402P, UGN0421P), XLATE=UGN0420T, X08300000
HOSTA=YES, PF4=4, PF9=4, PF7AND8=YES, UPDATE=YES, TYPE=RMV 08310000

* LICENSE CANCEL WARRANT EXEMPTION
PCTEG 'LWX', PGM=(UGN0430P, UGN0431P), XLATE=UGN0430T, X08330000
HOSTA=YES, PF4=4, PF9=4, PF7AND8=YES, UPDATE=YES, TYPE=RMV 08340000

* WARRANTS OFFENSE INQUIRY SCREEN
PCTEG 'LWX ', PGM=(UGN0440P, UGN0441P), XLATE=UGN0440T, X08360000
HOSTA=YES, PF4=4, PF9=4, TYPE=RMV 08370000
PCTEG 'LX ', PGM=(UGL0200P, UGL0201P), XLATE=UGL0200T, X08380000
UPDATE=YES, PF4=4, PF9=4, HOSTA=YES, PF7AND8=YES, TYPE=RMV 08390000
PCTEG 'LXS ', PGM=(UGL0300P, UGL0301P), XLATE=UGL0300T, X08400000
PF4=4, PF9=4, HOSTA=YES, PF7AND8=YES, TYPE=RMV 08410000

* COMMENTED OUT LXV, UGL0320 USED FOR PDPS
PCTEG 'LXV ', PGM=(UGL0320P, UGL0321P), XLATE=UGL0320T, 08420000

PCTEG 'LXV ', PGM=(UGL0320P, UGL0321P), XLATE=UGL0320T, 08430000
* PF4=4,PF9=4,HOSTA=Yes,PF7AND8=Yes,TYPE=RMV 08440000
* CDLIS INQUIRY DETAIL
  PCTEG 'LYD ',PGM=(UGL0292P,UGL0293P),XLATE=UGL0292T,
  UPDATE=Yes,HOSTA=Yes,TYPE=RMV 08450000
* CDLIS REQUEST HISTORY
  PCTEG 'LYH ',PGM=(UGL0280P,UGL0281P),XLATE=UGL0280T,
  UPDATE=Yes,PF4=4,PF9=4,HOSTA=Yes,TYPE=RMV 08470000
* ISSUE CDLIS REQUEST
  PCTEG 'LYI ',PGM=(UGL0270P,UGL0271P),XLATE=UGL0270T,
  UPDATE=Yes,PF4=4,PF9=4,HOSTA=Yes,TYPE=RMV 08500000
* CDLIS INQUIRY RESOLUTION
  PCTEG 'LYR ',PGM=(UGL0290P,UGL0291P),XLATE=UGL0290T,
  UPDATE=Yes,PF4=4,PF7AND8=YES,PF9=4,HOSTA=Yes,TYPE=RMV 08530000
* MATT MESSAGE DETAIL VIEWER
  PCTEG 'MATD',PGM=(UGT0040P,UGT0041P),XLATE=UGT0040T,
  UPDATE=Yes,PF7AND8=YES,HOSTA=Yes,TYPE=RMV 08560000
* MATT TEST CASE EDITOR
  PCTEG 'MATE',PGM=(UGT0010P,UGT0011P),XLATE=UGT0010T,
  UPDATE=Yes,PF7AND8=YES,HOSTA=Yes,TYPE=RMV 08590000
* MATT TEST CASE MANIPULATOR
  PCTEG 'MATM',PGM=(UGT0020P,UGT0021P),XLATE=UGT0020T,
  UPDATE=Yes,PF7AND8=YES,HOSTA=Yes,TYPE=RMV 08620000
* MATT MESSAGE VIEWER
  PCTEG 'MATV',PGM=(UGT0030P,UGT0031P),XLATE=UGT0030T,
  UPDATE=Yes,PF7AND8=YES,HOSTA=Yes,TYPE=RMV 08650000
* CUT OFF THE USE OF ALARS VERSIONS OF THESE PROGRAMS BECAUSE
  OF THE NEW OUT-OF-STATE REPORTING CHANGES BEING IMPLEMENTED
  PCTEG 'MDF ',PGM=(UGM0090P,UGM0091P),UPDATE=YES,
  XLATE=UGM0090T,HOSTA=YES,SCROLL=YES,PF9=4 08680000
  PCTEG 'MDH ',PGM=(UGM0092P,UGM0091P),UPDATE=YES,
  XLATE=UGM0090T,HOSTA=YES,SCROLL=YES,PF9=4 08710000
* TRANSFER ENTRY AS REQUESTED BY DANA
  PCTEG 'MIRR',PGM=UGZ0009P,TASK=MIRR,XFER=YES,TYPE=RMV 08740000
* MRB NO CREDIT PERIOD MAINTENANCE
  PCTEG 'MNC ',PGM=(UGM0012P,UGM0013P),UPDATE=YES,
  XLATE=UGM0012T,HOSTA=YES,SCROLL=NO,PF4=H,PF9=H 08770000
* TRANSFER ENTRY AS REQUESTED BY DANA
  PCTEG 'MODL',PGM=UGZ0009P,TASK=MODL,XFER=YES,TYPE=RMV 08800000
* MRB DUPLICATE LICENSE SCROLL
  PCTEG 'MRBS',PGM=(UGM0001P,UGM0002P),XLATE=UGM0001T,
  HOSTA=YES,SCROLL=YES,INTERNL=YES,DUPKEY=MRBX 08830000
* MRB CUSTOMER SERVICE DETAIL INPUT
  PCTEG 'MS ',PGM=(UGM0060P,UGM0061P),XLATE=UGM0060T,
  HOSTA=YES,UPDATE=YES,PF4=APEG,PF9=APEG 08860000
* MRB CUSTOMER SERVICE SUMMARY
  PCTEG 'MSC ',PGM=(UGM0066P,UGM0067P),XLATE=UGM0066T,
  HOSTA=YES,SCROLL=YES,PF4=W,PF9=W 08890000
* MRB CUSTOMER SERVICE DETAIL ENTRY (2)
  PCTEG 'MSP ',PGM=(UGM0064P,UGM0061P),XLATE=UGM0060T,
  HOSTA=YES,UPDATE=YES,PF4=W45,PF9=W45 08920000
* MRB CUSTOMER SERVICE REFERRAL
  PCTEG 'MSR ',PGM=(UGM0068P,UGM0069P),XLATE=UGM0068T,
  HOSTA=YES,PF4=W,PF9=W,SCROLL=YES 08950000
* MRB CUSTOMER SERVICE DETAIL UPDATE
  PCTEG 'MSU ',PGM=(UGM0062P,UGM0063P),XLATE=UGM0060T,
  HOSTA=YES,UPDATE=YES,PF4=W,PF9=W 08980000
* NMVTIS DUPLICATE VIN RESOLUTION ENTRY
  PCTEG 'NMV ',PGM=(UGM0066P,UGM0067P),XLATE=UGM0066T,
  HOSTA=YES,SCROLL=YES,PF4=W,PF9=W 09010000
* NMVTIS DUPLICATE VIN RESOLUTION ENTRY
  PCTEG 'NMV ',PGM=(UGM0066P,UGM0067P),XLATE=UGM0066T,
  HOSTA=YES,SCROLL=YES,PF4=W,PF9=W 09040000
<table>
<thead>
<tr>
<th>PCTEG</th>
<th>NMH - NMVTIS Messages Log Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>'NMD ', PGM=(UGR1351P, UGR1352P), XLATE=UGR1350T, UPDATE=YES, HOSTA=YES,</td>
<td>X09070000</td>
</tr>
<tr>
<td>* NMH - NMVTIS Messages Log Screen</td>
<td></td>
</tr>
<tr>
<td>'NMH ', PGM=(UGR1361P, UGR1362P), XLATE=UGR1360T, UPDATE=YES, HOSTA=YES, NULL9=NO, PF4=012, PF9=012, SCROLL=YES,</td>
<td>X09110000</td>
</tr>
<tr>
<td>* NMVTIS Problem Resolution Screen</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NMH ', PGM=(UGR1341P, UGR1342P), XLATE=UGR1340T, UPDATE=YES, HOSTA=YES,</td>
</tr>
<tr>
<td>* NEW PASSWORD (PASSWORD CHANGER)</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NPAS', PGM=(UGN0280P), XLATE=UGN0280T, HOSTA=YES, UPDATE=YES,</td>
</tr>
<tr>
<td>* NONRENEW, GENERAL ADDRESS EDIT</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRAE', PGM=(UGN0320P, UGN0321P, UGN0322P), XLATE=UGN0320T, HOSTA=YES, PF4=4, PF9=4, AEDNEW=NO,</td>
</tr>
<tr>
<td>* NONRENEW, GENERAL ADDRESS EDIT / 40 BYTE</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRBE', PGM=(UGN0340P, UGN0341P, UGN0342P), XLATE=UGN0340T, HOSTA=YES, PF4=4, PF9=4, AEDNEW=NO,</td>
</tr>
<tr>
<td>* NONRENEW, CLEAR TICKET/BILL</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRC ', PGM=(UGN0120P, UGN0121P), XLATE=UGN0120T, HOSTA=YES, UPDATE=YES,</td>
</tr>
<tr>
<td>* NONRENEW, DISPLAY TICKET CLEAR HISTORY BY CLERK</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRKH', PGM=(UGN0240P, UGN0241P), XLATE=UGN0240T, HOSTA=YES, SCROLL=YES, RESCROLL=YES,</td>
</tr>
<tr>
<td>* NONRENEW, INQUIRE FOR LICENSE</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRL ', PGM=(UGN0040P, UGN0041P), XLATE=UGN0040T, HOSTA=YES, PF4=4, PF9=4, PF7AND8=YES,</td>
</tr>
<tr>
<td>* NONRENEW, DISPLAY TICKET CLEAR HISTORY BY LICENSE</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRLH', PGM=(UGN0180P, UGN0181P), XLATE=UGN0180T, HOSTA=YES, PF4=4, PF9=4, PF7AND8=YES,</td>
</tr>
<tr>
<td>* NONRENEW, MARK</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRM ', PGM=(UGN0080P, UGN0081P), XLATE=UGN0080T, HOSTA=YES, PF4=0, PF9=0, UPDATE=YES,</td>
</tr>
<tr>
<td>* NONRENEW, MARK EXTERNAL (PERS ONLY)</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRMX', PGM=(UGN0300P, UGN0301P), XLATE=UGN0300T, HOSTA=YES, PF4=4, PF9=4, UPDATE=YES,</td>
</tr>
<tr>
<td>* NONRENEW, INQUIRE FOR REGISTRATION</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRR ', PGM=(UGN0020P, UGN0021P), XLATE=UGN0020T, HOSTA=YES, PF4=0, PF9=0, PF7AND8=YES,</td>
</tr>
<tr>
<td>* NONRENEW, DISPLAY TICKET CLEAR HISTORY BY REGISTRATION</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRRH', PGM=(UGN0200P, UGN0201P), XLATE=UGN0200T, HOSTA=YES, PF4=0, PF9=0, PF7AND8=YES,</td>
</tr>
<tr>
<td>* NONRENEW, DISPLAY SURROGATE AND OWNER FOR REG</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRRS', PGM=(UGN0460P, UGN0461P), XLATE=UGN0460T, HOSTA=YES, PF4=0, PF9=0,</td>
</tr>
<tr>
<td>* NONRENEW, DISPLAY SWAP TARGET</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRS ', PGM=(UGN0100P, UGN0101P), XLATE=UGN0100T, HOSTA=YES, PF4=0, PF9=0, UPDATE=YES,</td>
</tr>
<tr>
<td>* NONRENEW, INQUIRE FOR TICKET/BILL</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRT ', PGM=(UGN0060P, UGN0061P), XLATE=UGN0060T, HOSTA=YES,</td>
</tr>
<tr>
<td>* NONRENEW, DISPLAY TICKET CLEAR HISTORY BY TICKET/BILL</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRTBH', PGM=(UGN0220P, UGN0221P), XLATE=UGN0220T, HOSTA=YES, SCROLL=YES, RESCROLL=YES,</td>
</tr>
<tr>
<td>* NONRENEW, CHANGE TICKET/BILL NUMBER</td>
<td></td>
</tr>
<tr>
<td>PCTEG</td>
<td>'NRX ', PGM=(UGN0160P, UGN0161P), XLATE=UGN0160T,</td>
</tr>
</tbody>
</table>
HOSTA=YES, UPDATE=YES

* NONRENEW, CLEAR ALL TICKET/BILL IN TOWN
PCTEG 'NRZ ', PGM=(UGN0140P, UGN0141P), XLATE=UGN0140T,
HOSTA=YES, UPDATE=YES, PF4=04, PF9=04
X09680000

* NONRENEW, LOOKUP ZIPCODE FOR MASS TOWN/SUBURB
PCTEG 'NZIP', PGM=UGN0260P, XLATE=UGN0260T
9710000

* OVERWEIGHT REDUCIBLE PERMIT ISSUANCE SCREEN
PCTEG 'OW ', PGM=(UGW1111P, UGW1112P), XLATE=UGW1110T,
UPDATE=YES, PF4=01E, PF9=01E, HOSTA=YES, TYPE=RMV
X09730000

* OWH – OVERWEIGHT HISTORY
PCTEG 'OWH ', PGM=(UGW1131P, UGW1132P), XLATE=UGW1130T,
HOSTA=YES, PF4=E01, PF9=01, SCROLL=YES, TYPE=RMV
X09810000

* OWI – OVERWEIGHT INQUIRY
PCTEG 'OWI ', PGM=(UGW1141P, UGW1142P), XLATE=UGW1140T,
HOSTA=YES, PF4=E01, PF9=01, TYPE=RMV
X09820000

* TRANSFER ENTRY AS REQUESTED BY DANA
PCTEG 'PROD', PGM=UGZ0009P, TASK=PROD, XFER=YES, TYPE=RMV
X09850000
PCTEG 'RA ', PGM=(UGR1271P, UGR1272P), XLATE=UGR1270T,
UPDATE=YES, HOSTA=YES, PF4=012, PF9=012, HOSTA=YES
X09870000

* RBS – REGISTRATION BANK SCROLL
PCTEG 'RBS ', PGM=(UGR2201P, UGR2202P), XLATE=UGR2200T,
SCROLL=YES, RESCROLL=YES, HOSTA=YES, ALIAS='URBS'
X09910000

* REGISTRATION HISTORY
PCTEG 'RH ', PGM=(UGR1251P, UGR1252P), XLATE=UGR1250T,
SCROLL=YES, RESCROLL=YES, HOSTA=YES, PF4=01
X09940000
PCTEG 'RI ', PGM=(UGR5061P, UGR5062P), XLATE=UGR5060T,
HOSTA=YES, PF4=1024, PF9=1024, ALIAS='URI'
X09970000
PCTEG 'RI2 ', PGM=(UGR2061P, UGR2062P), XLATE=UGR2060T,
HOSTA=YES, PF4=1024, PF9=1024, HOSTA=YES
X10000000
PCTEG 'RI3 ', PGM=(UGR3061P, UGR3062P), XLATE=UGR3060T,
HOSTA=YES, PF4=1024, PF9=1024, HOSTA=YES
X10010000

* PCTEG 'RNF ', PGM=(UGR1231P, UGR1232P), XLATE=UGR1230T,
SCROLL=YES, RESCROLL=YES, HOSTA=YES, PF9=05, PF4=05
X10130000

* RP – VANITY/SPECIAL PLATE ORDER
PCTEG 'RP ', PGM=(UGR0311P, UGR0312P), XLATE=UGR0310T,
SCROLL=NO, UPDATE=YES, HOSTA=YES, TYPE=RMV
X10230000

* REG SUPERQUERY.
X10270000
PCTEG 'RSQ ', PGM=(UGR0020P, UGR0021P), HOSTA=YES, SCROLL=YES, RESCROLL=YES, X10280000
       XLATE=UGR0020T X10290000
10320000
PCTEG 'RVN ', PGM=(UGR2141P, UGR2142P), XLATE=UGR2140T, UPDATE=YES, PF4=01, PF9=01, HOSTA=YES, ALIAS='URVN' X10330000
       NULL9=YES, TASK=UG05 X10340000
10370000
*    PCTEG 'R1A ', PGM=(UGR1181P, UGR1182P), XLATE=UGR1180T, SCROLL=NO, UPDATE=YES, HOSTA=YES, PF7AND8=NO, INTERNL=YES, NULL9=YES, PF9=L, PF5=YES, TASK=UG05 X10390000
       NULL9=NO, PF9=L, PF5=YES X1040000
*    PCTEG 'R1B ', PGM=(UGR1191P, UGR1192P), XLATE=UGR1190T, SCROLL=NO, UPDATE=YES, HOSTA=YES, PF7AND8=NO, INTERNL=YES, NULL9=YES, PF9=L, PF5=YES, TASK=UG05 X10430000
       HOSTA=YES, PF9=045, TASK=UG05 X10440000
*    R1C - IN CUSTODY LESSEE INFORMATION X10460000
    * TRANSFER ENTRY AS REQUESTED BY DANA X10470000
PCTEG 'SCH', PGM=(UGS1101P, UGS1102P), XLATE=UGS1100T, PF4=4, PF9=4, HOSTA=YES, PF7AND8=NO, X10480000
       ALIAS='USH' X10490000
PCTEG 'SH ', PGM=(UGS3051P, UGS3052P), HOSTA=YES, SCROLL=YES, PF4=04, PF9=04, X1050000
       ALIAS='USH' X10510000
PCTEG 'SH1 ', PGM=(UGS0100P, UGS0101P), HOSTA=YES, PF7AND8=YES, PF4=4, PF9=4, TYPE=RMV X10520000
PCTEG 'SH2 ', PGM=(UGS1051P, UGS1052P), HOSTA=YES, PF7AND8=YES, PF4=4, PF9=4, TYPE=RMV X10530000
PCTEG 'S0A ', PGM=(UGS1171P, UGS1172P), XLATE=UGS1170T, UPDATE=YES, PF4=74, PF9=4, HOSTA=YES, PF7AND8=YES, TYPE=RMV X10540000
PCTEG 'SOC ', PGM=(UGS3151P, UGS3152P), XLATE=UGS3150T, UPDATE=YES, PF4=84, PF9=4, HOSTA=YES, PF7AND8=YES, TYPE=RMV X10550000
PCTEG 'SOC1', PGM=(UGS1151P, UGS1152P), XLATE=UGS1150T, UPDATE=YES, PF4=84, PF9=4, HOSTA=YES, PF7AND8=YES, TYPE=RMV X10560000
PCTEG 'SOW ', PGM=(UGS1161P, UGS1162P), XLATE=UGS1160T, UPDATE=YES, PF4=94, PF9=4, HOSTA=YES, PF7AND8=TYPE=RMV X10570000
*    TRANSFER ENTRY AS REQUESTED BY DANA X10580000
PCTEG 'SPCL', PGM=UGZ0009P, TASK=SPCL, XFER=YES, TYPE=RMV X10590000
PCTEG 'STAT', PGM=UGZ0010P, HOSTA=NO, X1060000
       FLAGS=FF00000000000000 X10610000
PCTEG 'SYSM', PGM=UGZ0009P, TASK=SYSM, XFER=YES, TYPE=RMV X10620000
*    TITLE AUDIT SCREEN X10630000
PCTEG 'TAS ', PGM=(UGR1081P, UGR1082P), XLATE=UGR1080T, UPDATE=YES, HOSTA=YES, NULL9=NO, PF7AND8=YES, TYPE=RMV X10640000
*    TRANSFER ENTRY X10650000
PCTEG 'TEST', PGM=UGZ0009P, TASK=TEST, XFER=YES, TYPE=RMV X10660000
PCTEG 'TLI ', PGM=(UGL0070P, UGL0071P), HOSTA=YES, PF4=4, PF9=4 X10670000
PCTEG 'TPA ', PGM=(UGU2121P, UGU2122P), XLATE=UGU2120T, UPDATE=YES, PF4=6, PF9=0146, HOSTA=YES X10680000
PCTEG 'TPH ', PGM=(UGU2041P, UGU2042P), XLATE=UGU2040T, UPDATE=YES, HOSTA=YES, PF9=4, PF7AND8=YES, PP4=4 X10690000
PCTEG 'TPIC', PGM=(UGU3021P, UGU3022P), XLATE=UGU3020T, X10700000

Appendix B – Guest Process Control Table
**Appendix B – Guest Process Control Table**

```
UPDATE=YES,PF9=01,HOSTA=YES,SCROLL=YES,INTERNL=YES       11030000
PCTEG 'TPOI',PGM=(UGU2011P,UGU2012P),XDATE=UGU2010T,       X11040000
HOSTA=YES,PF9=4                                           11050000
PCTEG 'TPTH',PGM=(UGU1131P,UGU1132P),XDATE=UGU1130T,       X11060000
HOSTA=YES,PF9=6,PF7AND8=YES,PF4=6                         11070000

* TRBS - REGISTRATION BANK SCROLL

PCTEG 'TRBS',PGM=(UGR1201P,UGR1202P),XDATE=UGR1200T,       X11090000
SCROLL=YES,RESCROLL=YES,HOSTA=YES                         11110000
PCTEG 'TRI ',PGM=(UGR4061P,UGR4062P),XDATE=UGR4060T,       X11110000
HOSTA=YES,PF4=1024,PF9=1024                               11112000
PCTEG 'TRSN',PGM=(UGRI102P,UGRO102P),XDATE=UGRX102T,       X11130000
SCROLL=YES,RESCROLL=YES,HOSTA=YES                         11114000
PCTEG 'TRSR',PGM=(UGRI100P,UGRO100P),XDATE=UGRX100T,       X11150000
SCROLL=YES,RESCROLL=YES,DUPKEY=UR02,HOSTA=YES             11116000
PCTEG 'TR1A',PGM=(UGR1181P,UGR1182P),XDATE=UGR1180T,       X11170000
SCROLL=NO,UPDATE=YES,PF9=145,HOSTA=YES,NULL9=YES          11118000
PCTEG 'TR1B',PGM=(UGR1191P,UGR1192P),XDATE=UGR1190T,       X11190000
SCROLL=NO,UPDATE=YES,HOSTA=YES,PF7AND8=YES,INTERNL=YES,   X11200000
NULL9=YES,PF9=L,PF5=YES                                    11121000
PCTEG 'TSH ',PGM=(UGS2051P,UGS2052P),HOSTA=YES,SCROLL=YES, X11220000
PF4=4,PF9=4                                              11125000
PCTEG 'TVH ',PGM=(UGU2111P,UGU2112P),XDATE=UGU2110T,       X11280000
HOSTA=YES,PF7AND8=YES,PF4=01,PF9=01                      11130000

* OLD RVN ENTRY

PCTEG 'TVN ',PGM=(UGR1141P,UGR1142P),XDATE=UGR1140T,       X11300000
UPDATE=YES,PF4=01,PF9=01,HOSTA=YES,ALIAS='UTVN'          11131000

* OLD VT PCTEG ENTRY

PCTEG 'TVT ',PGM=(UGR2261P,UGR2262P),XDATE=UGR2260T,       X11410000
SCROLL=YES,RESCROLL=YES,HOSTA=YES,PF4=1,PF9=1             11132000
PCTEG 'TVT1',PGM=(UGR1261P,UGR1262P),XDATE=UGR1260T,       X11450000
SCROLL=YES,RESCROLL=YES,HOSTA=YES,PF4=1,PF9=1             CP1294 11134000
PCTEG 'TXH ',PGM=(UGR1281P,UGR1282P),XDATE=UGR1280T,       X11470000
SCROLL=YES,RESCROLL=YES,HOSTA=YES,PF4=2,PF9=1             CP0695 11137000
PCTEG 'TXH ',PGM=(UGS2151P,UGS2152P),HOSTA=YES,SCROLL=YES, X11500000
PF4=4,PF9=4                                              CP0695 11138000
PCTEG 'TSH ',PGM=(UGS2051P,UGS2052P),HOSTA=YES,SCROLL=YES, X11600000
PF4=4,PF9=4                                              CP1294 11139000
PCTEG 'T1A ',PGM=(UGR1211P,UGR1212P),XDATE=UGR1210T,       X11610000
UPDATE=YES,HOSTA=YES,PF4=012,PF9=01L,NULL9=YES,TYPE=RMV  11140000
PCTEG 'T1A ',PGM=(UGR1211P,UGR1212P),XDATE=UGR1210T,       X11620000
UPDATE=YES,HOSTA=YES,PF4=210,PF9=01L,NULL9=YES,TYPE=RMV  11141000
PCTEG 'T1B ',PGM=(UGR1221P,UGR1222P),XDATE=UGR1220T,       X11630000
UPDATE=YES,HOSTA=YES,INTERNL=YES,TYPE=RMV               11142000
PCTEG 'T1B ',PGM=(UGR1221P,UGR1222P),XDATE=UGR1220T,       X11640000
UPDATE=YES,HOSTA=YES,INTERNL=YES,TYPE=RMV               11143000

* END OF ADDITIONS FOR EDS
```

---

**Changes for EDS 11/13/92 Deb V.**

```
* PCTEG 'T1A ',PGM=(UGR1211P,UGR1212P),XDATE=UGR1210T,       X11530000
  UPDATE=YES,HOSTA=YES,PF4=012,PF9=01L,NULL9=YES,TYPE=RMV  11150000
* PCTEG 'T1A ',PGM=(UGR1211P,UGR1212P),XDATE=UGR1210T,       X11540000
  UPDATE=YES,HOSTA=YES,PF4=012,PF9=01L,NULL9=YES,TYPE=RMV  11151000
* END OF 11/13/92 CHANGE
* END OF ADDITIONS FOR EDS
```
**Registry of Motor Vehicles – UMS Programmer’s Manual**

**Appendix B – Guest Process Control Table**

* TIC – ALTERNATE ADDRESS GUEST PROGRAMS CP1098
  PCTEG 'TIC ', PGM=(UGR1311P,UGR1312P), XLATE=UGR1310T,
  SCROLL=NO, UPDATE=YES, HOSTA=YES, PF4=012, PF9=01,
  NULL9=YES
* PREV
*
* IMAGING DUPLICATE LICENSE OR MASS ID OR LIQUOR ID
  PCTEG 'UDUP', PGM=(UGL0640P, UGL0641P), XLATE=UGL0640T,
  PF4=4, HOSTA=YES, UPDATE=YES
  PCTEG 'UHMT', PGM=UGZ0009P, TASK=UHMT, XFER=YES, TYPE=RMV
  PCTEG 'UL ', PGM=UGZ0010P, HOSTA=NO
* IMAGING CUSTOMER ADD SCREEN
  PCTEG 'ULC ', PGM=(UGL0540P, UGL0541P), XLATE=UGL0540T,
  UPDATE=YES, HOSTA=YES, NULL9=YES, PF9=4, PF4=4,
  ALIAS='LC '
* MRB AT-FAULT CLAIM INQUIRY
  PCTEG 'UMA ', PGM=(UGM0030P, UGM0031P), XLATE=UGM0030T,
  HOSTA=YES, PF4=F, PF9=F4, UPDATE=YES, PF7AND8=YES,
  ALIAS='MA '
* MRB BOARD OF APPEALS REVERSAL
  PCTEG 'UMBA', PGM=(UGM0032P, UGM0033P), XLATE=UGM0032T,
  ALIAS='MBA ', HOSTA=YES, PF4=F, PF9=F4, UPDATE=YES, PF7AND8=YES
* MRB PENDING BATCH REVIEW
  PCTEG 'UMBP', PGM=(UGM0152P, UGM0153P), UPDATE=NO,
  XLATE=UGM0152T, HOSTA=YES, SCROLL=YES
* MRB COMPREHENSIVE CLAIM INQUIRY
  PCTEG 'UMC ', PGM=(UGM0040P, UGM0041P), XLATE=UGM0040T,
  HOSTA=YES, PF4=G, PF9=G4, UPDATE=YES, PF7AND8=YES,
  ALIAS='MC '
* MRB ADMINISTRATIVE DETAIL
  PCTEG 'UMDD', PGM=(UGM0094P, UGM0095P), UPDATE=NO,
  XLATE=UGM0094T, HOSTA=YES, SCROLL=NO, PF4=4,
  ALIAS='MDD '
* MRB DRIVING HISTORY
  PCTEG 'UMDF', PGM=(UGM0090P, UGM0091P), UPDATE=NO,
  XLATE=UGM0090T, HOSTA=YES, SCROLL=YES, PF4=9
  PCTEG 'UMDH', PGM=(UGM0092P, UGM0091P), UPDATE=NO,
  XLATE=UGM0092T, HOSTA=YES, SCROLL=YES, PF4=4
* MRB SDIP INQUIRY MENU
  PCTEG 'UMI ', PGM=(UGM0022P), XLATE=UGM0022T, HOSTA=NO
* MRB SDIP INQUIRY OUTPUT SCREEN
  PCTEG 'UMIQ', PGM=(UGM0021P), XLATE=UGM0021T, HOSTA=YES,
  PF7AND8=YES, INTERNL=YES, UPDATE=YES
* MRB SDIP INQUIRY NEW POLICY

---

*IMAGING DUPLICATE LICENSE OR MASS ID OR LIQUOR ID*  
PCTEG 'UDUP', PGM=(UGL0640P, UGL0641P), XLATE=UGL0640T,  
PF4=4, HOSTA=YES, UPDATE=YES  
PCTEG 'UHMT', PGM=UGZ0009P, TASK=UHMT, XFER=YES, TYPE=RMV  
PCTEG 'UL ', PGM=UGZ0010P, HOSTA=NO  

*IMAGING CUSTOMER ADD SCREEN*  
PCTEG 'ULC ', PGM=(UGL0540P, UGL0541P), XLATE=UGL0540T,  
UPDATE=YES, HOSTA=YES, NULL9=YES, PF9=4, PF4=4,  
ALIAS='LC '  

*MRB AT-FAULT CLAIM INQUIRY*  
PCTEG 'UMA ', PGM=(UGM0030P, UGM0031P), XLATE=UGM0030T,  
HOSTA=YES, PF4=F, PF9=F4, UPDATE=YES, PF7AND8=YES,  
ALIAS='MA '  

*MRB BOARD OF APPEALS REVERSAL*  
PCTEG 'UMBA', PGM=(UGM0032P, UGM0033P), XLATE=UGM0032T,  
ALIAS='MBA ', HOSTA=YES, PF4=F, PF9=F4, UPDATE=YES, PF7AND8=YES  

*MRB PENDING BATCH REVIEW*  
PCTEG 'UMBP', PGM=(UGM0152P, UGM0153P), UPDATE=NO,  
XLATE=UGM0152T, HOSTA=YES, SCROLL=YES  

*MRB COMPREHENSIVE CLAIM INQUIRY*  
PCTEG 'UMC ', PGM=(UGM0040P, UGM0041P), XLATE=UGM0040T,  
HOSTA=YES, PF4=G, PF9=G4, UPDATE=YES, PF7AND8=YES,  
ALIAS='MC '  

*MRB ADMINISTRATIVE DETAIL*  
PCTEG 'UMDD', PGM=(UGM0094P, UGM0095P), UPDATE=NO,  
XLATE=UGM0094T, HOSTA=YES, SCROLL=NO, PF4=4,  
ALIAS='MDD '  

*MRB DRIVING HISTORY*  
PCTEG 'UMDF', PGM=(UGM0090P, UGM0091P), UPDATE=NO,  
XLATE=UGM0090T, HOSTA=YES, SCROLL=YES, PF4=9  
PCTEG 'UMDH', PGM=(UGM0092P, UGM0091P), UPDATE=NO,  
XLATE=UGM0092T, HOSTA=YES, SCROLL=YES, PF4=4  

*MRB SDIP INQUIRY MENU*  
PCTEG 'UMI ', PGM=(UGM0022P), XLATE=UGM0022T, HOSTA=NO  

*MRB SDIP INQUIRY OUTPUT SCREEN*  
PCTEG 'UMIQ', PGM=(UGM0021P), XLATE=UGM0021T, HOSTA=YES,  
PF7AND8=YES, INTERNL=YES, UPDATE=YES  

*MRB SDIP INQUIRY NEW POLICY*
PCTEG 'UMI1', PGM=(UGM0023P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, UPDATE=YES, PF4=E4, PF9=E4
* MRB SDIP INQUIRY POLICY RENEWAL
PCTEG 'UMI2', PGM=(UGM0024P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, UPDATE=YES, PF4=E4, PF9=E4
* MRB SDIP INQUIRY ADD OPERATOR
PCTEG 'UMI3', PGM=(UGM0025P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, UPDATE=YES, PF4=E4, PF9=E4
* MRB SDIP INQUIRY ADD COLLISION
PCTEG 'UMI4', PGM=(UGM0026P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, UPDATE=YES, PF4=E4, PF9=E4
* MRB SDIP INQUIRY ADD LIABILITY
PCTEG 'UMI5', PGM=(UGM0027P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, UPDATE=YES, PF4=E4, PF9=E4
* MRB SDIP INQUIRY REINSTATEMENT
PCTEG 'UMI6', PGM=(UGM0028P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, UPDATE=YES, PF4=E4, PF9=E4
* MRB SDIP INQUIRY OPERATOR INPUT SCREEN
PCTEG 'UMI9', PGM=(UGM0020P, UGM0029P), XLATE=UGM0020T, HOSTA=YES, PF4=E4, PF9=E4, ALIAS='MI9'
* MRB LATE APPEAL GENERATION
PCTEG 'UMLA', PGM=(UGM0080P, UGM0081P), XLATE=UGM0080T, HOSTA=YES, UPDATE=YES, PF4=F, PF9=F, ALIAS='MLA'
PCTEG 'UMM ', PGM=UGZ0010P, HOSTA=NO
* MRB OPERATOR SUMMARY DISPLAY
PCTEG 'UMO ', PGM=(UGM0010P, UGM0011P), XLATE=UGM0010T, HOSTA=YES, SCROLL=YES, PF4=45, PF9=45, ALIAS='MO'
* MRB NOWN SUMMARY DISPLAY
PCTEG 'UMON', PGM=(UGM0070P, UGM0071P), XLATE=UGM0070T, HOSTA=YES, SCROLL=YES, PF4=5, PF9=5, ALIAS='MON'
* MRB ADD CITATION
PCTEG 'UMV ', PGM=(UGM0250P, UGM0251P), XLATE=UGM0051T, HOSTA=YES, PF4=A, PF9=A, UPDATE=YES
* MRB ADD CITATION
PCTEG 'UMVH', PGM=(UGM0052P, UGM0053P), XLATE=UGM0052T, HOSTA=YES, PF4=A, PF9=A, UPDATE=YES
* MRB CITATION INQUIRY
PCTEG 'UMVI', PGM=(UGM0050P, UGM0051P), XLATE=UGM0050T, HOSTA=YES, PF4=A, PF9=A, SCROLL=YES, ALIAS='MVI'
* MRB DUPLICATE CITATION SCROLL
PCTEG 'UMVS', PGM=(UGM0055P, UGM0056P), XLATE=UGM0055T, HOSTA=YES, SCROLL=YES, INTERNL=YES, DUPKEY=UMVX
* MRB SUSPENDED CITATIONS
PCTEG 'UMUV', PGM=(UGM0252P, UGM0253P), XLATE=UGM0051T, HOSTA=YES, PF4=A, PF9=A, UPDATE=YES
* MRB CITATIONS - MODIFY VIOLATOR INFORMATION
PCTEG 'UMV1', PGM=(UGM0254P, UGM0255P), XLATE=UGM0051T, HOSTA=YES, PF4=A, PF9=A, UPDATE=YES
* MRB CITATIONS - MODIFY VIOLATION INFORMATION
PCTEG 'UMV2', PGM=(UGM0256P, UGM0257P), XLATE=UGM0051T, HOSTA=YES, PF4=A, PF9=A, UPDATE=YES
* MRB CITATIONS - REVERSAL
PCTEG 'UMV3', PGM=(UGM0258P, UGM0259P), XLATE=UGM0051T, HOSTA=YES, PF4=A, PF9=A, UPDATE=YES
* MRB CITATIONS - MODIFY POLICE/LOCATION INFORMATION
PCTEG 'UMV4', PGM=(UGM0260P, UGM0261P), XLATE=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12830000
* MRB CITATIONS - REINSTATE
PCTEG 'UMV5',PGM=(UGM0262P,UGM0263P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12850000
* MRB CITATIONS - NULL PROCESS
PCTEG 'UMV6',PGM=(UGM0264P,UGM0265P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12860000
* MRB CITATIONS - HEARING REQUEST
PCTEG 'UMV7',PGM=(UGM0266P,UGM0267P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12870000
* MRB CITATIONS - COURT RESPONSE TO ORIGINAL HEARING
PCTEG 'UMV8',PGM=(UGM0272P,UGM0273P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12880000
* MRB CITATIONS - CHANGE CITATION NUMBER
PCTEG 'UMV9',PGM=(UGM0276P,UGM0277P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12890000
* MRB CITATIONS - CHANGE HEARING REQUEST
PCTEG 'UM7C',PGM=(UGM0269P,UGM0267P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12900000
* MRB CITATIONS - LATE HEARING REQUEST
PCTEG 'UM7L',PGM=(UGM0268P,UGM0267P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12910000
* MRB CITATIONS - NULL HEARING REQUEST
PCTEG 'UM7N',PGM=(UGM0270P,UGM0267P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12920000
* MRB CITATIONS - COURT RESPONSE TO CHANGED HEARING
PCTEG 'UM8L',PGM=(UGM0274P,UGM0273P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12930000
* MRB CITATIONS - NULL COURT RESPONSE
PCTEG 'UM8N',PGM=(UGM0275P,UGM0273P),Xlate=UGM0051T,
HOSTA=YES,PF4=A,PF9=A,UPDATE=YES  12940000
* MRB CITATIONS - CHANGE CITATION NUMBER
PCTEG 'UP',PGM=UGZ0010P,HOSTA=NO  13100000
PCTEG 'UPA',PGM=(UGU3121P,UGU3122P),Xlate=UGU3120T,
UPDATE=YES,PF4=6,PF9=45,HOSTA=YES  13110000
PCTEG 'UPH',PGM=(UGU3041P,UGU3042P),Xlate=UGU3040T,
HOSTA=YES,PF9=45,SCROLL=YES,PF4=45  13120000
PCTEG 'UPIC',PGM=(UGU4021P,UGU4022P),Xlate=UGU4020T,
UPDATE=YES,PF9=01,HOSTA=YES,SCROLL=YES,INTERNL=YES  13130000
* UPMV - MULTIPLE VEHICLE AMEND SCREEN
PCTEG 'UPMV',PGM=(UGU1141P,UGU1142P),Xlate=UGU1140T,
SCROLL=YES,UPDATE=YES,HOSTA=YES,PF4=6,PF9=01  13140000
PCTEG 'UPOI',PGM=(UGU3011P,UGU3012P),Xlate=UGU3010T,
HOSTA=YES,PF9=45  13150000
PCTEG 'UPTH',PGM=(UGU2131P,UGU2132P),Xlate=UGU2130T,
HOSTA=YES,PF4=6,PF7AND8=8,PF4=6  13160000
PCTEG 'UR',PGM=UGZ0010P,HOSTA=NO  13170000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,DUPKEY=BS01  13180000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,HOSA=YES,PF9=L,PF4=L  13190000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,HOSA=YES,PF9=L,PF4=L  13200000
PCTEG 'URN',PGM=(UGR0010P,UGR0011P),HOSTA=YES,SCROLL=YES,
RESCROLL=YES  13210000
PCTEG 'URSN',PGM=(UGR1091P,UGR1092P),Xlate=UGR1090T,
X13500000
RESCEOLL=YES  13220000
PCTEG 'USR',PGM=UGZ0010P,HOSTA=NO  13230000
PCTEG 'USRBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,HOSA=YES,PF4=6,PF9=01  13240000
PCTEG 'USPMV',PGM=(UGU1141P,UGU1142P),Xlate=UGU1140T,
SCROLL=YES,UPDATE=YES,HOSTA=YES,PF4=6,PF9=01  13250000
PCTEG 'USPOI',PGM=(UGU3011P,UGU3012P),Xlate=UGU3010T,
HOSTA=YES,PF9=45  13260000
PCTEG 'USPOI',PGM=(UGU3011P,UGU3012P),Xlate=UGU3010T,
HOSTA=YES,PF9=45  13270000
PCTEG 'UPH',PGM=(UGU3041P,UGU3042P),Xlate=UGU3040T,
HOSTA=YES,PF9=45,SCROLL=YES,PF4=45  13280000
PCTEG 'UPIC',PGM=(UGU4021P,UGU4022P),Xlate=UGU4020T,
UPDATE=YES,PF9=01,HOSTA=YES,SCROLL=YES,INTERNL=YES  13290000
* UPMV - MULTIPLE VEHICLE AMEND SCREEN
PCTEG 'UPMV',PGM=(UGU1141P,UGU1142P),Xlate=UGU1140T,
SCROLL=YES,UPDATE=YES,HOSTA=YES,PF4=6,PF9=01  13300000
PCTEG 'UPOI',PGM=(UGU3011P,UGU3012P),Xlate=UGU3010T,
HOSTA=YES,PF9=45  13310000
PCTEG 'UPTH',PGM=(UGU2131P,UGU2132P),Xlate=UGU2130T,
HOSTA=YES,PF4=6,PF7AND8=8,PF4=6  13320000
PCTEG 'UR',PGM=UGZ0010P,HOSTA=NO  13330000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,DUPKEY=BS01  13340000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,DUPKEY=BS01  13350000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,DUPKEY=BS01  13360000
PCTEG 'URBS',PGM=(UGR1201P,UGR1202P),Xlate=UGR1200T,
SCROLL=YES,RESCROLL=YES,DUPKEY=BS01  13370000
PCTEG 'URN',PGM=(UGR0010P,UGR0011P),HOSTA=YES,SCROLL=YES,
RESCROLL=YES  13380000
PCTEG 'URSN',PGM=(UGR1091P,UGR1092P),Xlate=UGR1090T,
SCROLL=YES, RESCROLL=YES, HOSTA=YES
PCTEG 'URSR', PGM=(UGR1241P, UGR1242P), XLATE=UGR1240T,
SCROLL=YES, RESCROLL=YES, DUPKEY=UR05, HOSTA=YES
13530000
PCTEG 'URSV', PGM=(UGRI101P, UGRO101P), XLATE=UGRX101T,
SCROLL=YES, RESCROLL=YES, DUPKEY=UR03, HOSTA=YES
13570000
* PCTEG 'URVN', PGM=(UGR1141P, UGR1142P), XLATE=UGR1140T,
   UPDATE=YES, PF4=01, PF9=01, HOSTA=YES
13620000
* PCTEG 'USH ', PGM=(UGS2051P, UGS2052P), HOSTA=YES, SCROLL=YES,
   PF4=04, PF9=04, TYPE=RMV, ALIAS='SH '
13690000
PCTEG 'UVH ', PGM=(UGU3111P, UGU3112P), XLATE=UGU3110T,
HOSTA=YES, SCROLL=YES, PF4=01, PF9=01
13700000
PCTEG 'VER ', PGM=UGZ0032P, HOSTA=NO
13780000
* NEW VT PCTEG ENTRY
PCTEG 'VT ', PGM=(UGR3261P, UGR3262P), XLATE=UGR3260T,
SCROLL=YES, RESCROLL=YES, HOSTA=YES, PF4=1, PF9=1
13860000
* NEW ZLEB ENTRY FOR IMAGING
PCTEG 'ZLEB', PGM=(UGL0040P)
13880000
* NEW ZL01 ENTRY FOR IMAGING
PCTEG 'ZL01', PGM=(UGL0500P), XLATE=UGL0500T,
PF4=4, PF9=4, NULL9=YES
13910000
* NEW ZN01 ENTRY FOR IMAGING
PCTEG 'ZN01', PGM=(UGL0510P), XLATE=UGL0510T,
PF4=5, PF9=5, NULL9=YES
13940000
AIF ('&MODEFLG' EQ 'P').AA040
13950000
LAST DC XL4'FFFFFFFF'
13960000
UGZ0002Q CSECT
13970000
XFER2 EQU * FIRST XFER ENTRY
13980000
Appendix C:

Host Process Control Table

The prototype of the macro is repeated here for convenience of reference. For a discussion of the parameters, please turn to Chapter 5.

PCTEH &NAME,&PGM=,&TASK=,&RMV=,&FLAGS=,&CLEAN=,
&DUPKEY=,&SERVICE=

NOTE: SPECIFICATION OF A NONEXISTANT ENTRY WILL RESULT IN UNDEFINED SYMBOLS OF THE FORM BTC$XXXX WHERE XXXX IS THE BLOCK TYPE CODE

This is the table itself with the current release entries. The start and finish of the table is provided for structural understanding.

UHZ0002P CSECT 02760000
NUMENT DC F'0' NUMBER OF ENTRIES 02770000
DC A(FIRST) POINT TO 1ST ENTRY 02780000
SERVICE 1,PGM=(UHRSTTLS,UHRSREGS) 02790000
SERVICE 2,PGM=(UHUC101P,UHUSSDIP,UHL0050P) 02800000
SERVICE 3,PGM=(UHRSREGS,UHL0050P) 02810000
* NDR/ESI ROUTING TABLE:
SERVICE 4,PGM=(UHL0200P,UHL0440P,UHLNDR2P) 02820000
SERVICE 5,PGM=(UHU101CP,MMRSSDIP,UHL0055P) 02830000
SERVICE 6,PGM=(UHL0055P,UHL0065P) 02840000
SERVICE 7,PGM=(UHU201CP,MMRSSDIP,UHL0055P) 02850000
* EXAM BOOKING TABLES AND CASH POSTING
SERVICE 8,PGM=(UHB0420P,UHB0400P,UHL0910P) 02860000
SERVICE 9,PGM=(UHL0055P,UHRSREGS) 02870000
*** SERVICE 11,PGM=(UHB0420P,UHB0400P,UHL0910P,UHB0700P) 02890000
SERVICE 12,PGM=(UHL0120P,UHL0120P,UHL0231P) 02910000
* IMAGE SERVER FUNCTIONS

** SERVICE 13, PGM=(UHLIMG1P, UHL0200P, UHL0120P, UHL0065P, UHL0910P, UX02990000, HL1000P, UHL0930P) 02980000
** SERVICE 14, PGM=(UHLIMG1P, UHL0200P) 03010000
** SERVICE 16, PGM=(UHL0065P, UHB0020P, UHL0910P, UHL0120P, UHL0055P, UX03040000, HL0234P) 03050000
** SERVICE 17, PGM=(UHL0045P) 03060000
** SERVICE 18, PGM=(UHL0214P, UHL0910P, UHL0217P) 03070000
** SERVICE 20, PGM=(MCCP996M, UHL0910P, UHL0120P, UHL0140P, UX03100000, UHL0120P, UHLNDR1P) 03110000
** SERVICE 23, PGM=(MCCPITEM) 03170000
** SERVICE 24, PGM=(UHRSTTLS, UHRSREGS, UHL0055P) 03180000
** SERVICE 25, PGM=(UHL0910P, MCCP991M, MSUSREIN, MRESPRNT) 03190000
** SERVICE 26, PGM=(UHLNDR1P, UHL0910P, UHL0920P, UHL0930P) 03200000
* MAB SURRENDER UPDATE
** SERVICE 27, PGM=(UHL0470P, UHL0120P, UHL0065P, UHL0130P, UX03220000, HL0400P) 03230000
* HANDICAP PLACARD UPDATE
** SERVICE 28, PGM=(UHL0670P, UHL0910P) 03260000
** SERVICE 29, PGM=(UHU301CP, MMRSSDIP, UHL0055P) 03270000
** SERVICE 30, PGM=(UHL0420P, UHL0400P) 03280000
* THIS SERVICE SET IS RESERVED FOR IMAGE TIMED EVENT PROCESSING
** SERVICE 32, PGM=(UHR0560T, UHR0580P) 03340000
*** SERVICE 33, PGM=(UHL0236P, UHL0910P, UHB0550P, UHL1120P) 03350000
*** SERVICE 33, PGM=(UHL0236P, UHB0550P, UHL0910P, UHL1120P) 03360000
** SERVICE 34, PGM=(UHL0208P, UHB0020P) 03390000
** SERVICE 35, PGM=(UHA0002P, UHA0006P, UHB0020P) 03400000
** SERVICE 36, PGM=(UHB2020P) 03410000
** SERVICE 37, PGM=(UHL1090T, UHL1090P, UHN0160P) 03420000
** SERVICE 38, PGM=(UHL0208P, UHL0214P, UHL0930P, UHB0020P) 03430000
** SERVICE 39, PGM=(UHL0208P, UHB0020P, UHL0910P, UHL0930P) 03440000
* MAB - MEDICAL AFFAIRS BRANCH
** SERVICE 41, PGM=(UHRSREGS) 03500000
* MAB - MEDICAL AFFAIRS BRANCH
** SERVICE 44, PGM=(UHH0470P) 03540000
** SERVICE 44, PGM=(UHH0470P, UHH0510P, UHH0350P, UHH0290P, UHL0670P, UX03560000

Appendix C – Host Process Control Table

238
HH0530P) 03570000
* 03580000
* NRT - NEW ROAD TEST PHASE 3 03590000
    HL0910P, UHL0930P) 03610000
* 03620000
* MAB - MEDICAL AFFAIRS BRANCH 03630000
  SERVICE 46, PGM=(UHH0620P, UHH0650P) 03640000
* 03650000
* NRT - ISSUE LICENSE FOR MOTORCYCLE COURSE 03660000
  SERVICE 47, PGM=(UHL0236P, UHL0910P, UHL0910P, UHL0216P) 03670000
* 03680000
  SERVICE 48, PGM=(UHL0208P, UHL0910P, UHL0120P, UHL0214P, UHL0910P, UX03690000
    HL0217P, UHB0020P, UHL0290P) 03700000
* ACTIVITY HOLD 03710000
  SERVICE 49, PGM=(UHL0790P, UHL0990P) 03720000
    HL0120P, UHLNDRP) 03740000
  SERVICE 51, PGM=(UHL0150P, UHL0120P, UHL0470P, UHL0229P, UHL0065P, UX03750000
    HL0055P, UHL0400P) 03760000
  SERVICE 52, PGM=(UHN0310P) 03770000
* 03780000
  AV01 - ADDRESS VALIDATION 03790000
    PCTEH AV01, PGM=(UHL1090P), SERVICE=37 03800000
* 03810000
  AV02 - ADDRESS REFORMAT 03820000
    PCTEH AV02, PGM=(UHL1095P), SERVICE=37 03830000
* 03840000
  BH45 - EXAM BOOKING - EXTRACT ROAD TEST SITES FOR BKCL 03850000
    PCTEH BH45, PGM=(UHB0450P), SERVICE=36 03860000
* 03870000
  BH46 - EXAM BOOKING - EXTRACT CLOSE OR REOPEN TEST SITES 03880000
    PCTEH BH46, PGM=(UHB0460P), RMV=BKCL, SERVICE=11 03890000
* 03900000
  BH47 - EXAM BOOKING - VALIDATE DRIVING SCHOOL FID 03910000
    PCTEH BH47, PGM=(UHB0470P), RMV=BKBB, SERVICE=36 03920000
* 03930000
  BH48 - EXAM BOOKING - DRIVING SCHOOL BLOCK BOOKING 03940000
    PCTEH BH48, PGM=(UHB0480P), RMV=BKBB, SERVICE=11 03950000
* 03960000
  BH49 - EXAM BOOKING - EXTRACT ROAD TEST TIMES FOR BKCL 03970000
    PCTEH BH49, PGM=(UHB049OP), RMV=BKCL 03980000
* 03990000
  BH55 - EXAM BOOKING - VALIDATE CONFIRMATION NUMBER 04000000
    PCTEH BH55, PGM=(UHB0550P) 98 04010000
  BH65 - EXAM BOOKING - DETERMINE BOOKING/CANCEL OPTIONS 04020000
    PCTEH BH65, PGM=(UHB0650P), SERVICE=16 04030000
* 04040000
  PCTEH BH75, PGM=(UHB0750P), RMV=ED, SERVICE=10 04050000
  BH85 - EXAM BOOKING - ISSUE LICENSE APPL & BOOK ROAD TEST 04060000
*** PCTEH BH85, PGM=(UHB0850P), SERVICE=33 04070000
  PCTEH BH85, PGM=(UHB0850P), SERVICE=11 04080000
*** PCTEH BH85, PGM=(UHB0850P), SERVICE=40 04090000
  BH86 - EXAM BOOKING - CANCEL ROAD TEST 04100000
    PCTEH BH86, PGM=(UHB0860P), RMV=BK01, SERVICE=33 04110000
  BH87 - EXAM BOOKING - MARK ALL ROAD TEST PAID 04120000
    PCTEH BH87, PGM=(UHB0870P), RMV=BK01, SERVICE=38 04130000

Appendix C – Host Process Control Table 239
| BH90 - EXAM BOOKING - BUILD ROAD EXAMS SCHEDULE | 04140000 |
| BH90 - EXAM BOOKING - UPDATE EXAM HISTORY FILE | 04170000 |
| BH91 - EXAM BOOKING - BUILD ROAD EXAMS SCHEDULE / UPDATE EXAM HISTORY | 04190000 |
| BH94 - EXAM BOOKING - BUILD ROAD EXAMS SCHEDULE / UPDATE EXAM HISTORY | 04200000 |
| BH95 - EXAM BOOKING - EXTRACT LICENSING FEES | 04210000 |
| BH97 - GRADE EXAM - CASH POSTING | 04220000 |
| URBS - UMS REGISTRATION BANK SCROLL | 04250000 |
| RBS - UMS REGISTRATION BANK SCROLL | 04270000 |
| UCCR CREDIT CARD RECONCILIATION | 04290000 |
| BAD PAYMENT HISTORY INQUIRY BY LIC | 04310000 |
| BAD PAYMENT HISTORY INQUIRY BY FID | 04340000 |
| BAD PAYMENT HISTORY INQUIRY BY REG | 04360000 |
| BAD PAYMENT HISTORY INQUIRY BY TITIE | 04390000 |
| BAD PAYMENT HISTORY INQUIRY BY PLATE ORDER (PLTO) | 04410000 |
| COR - OBLIGATION RESOLUTION INQUIRY BY REG | 04430000 |
| BAD PAYMENT HISTORY INQUIRY BY LICENSE | 04450000 |
| OBLIGATION RESOLUTION DETAILS | 04480000 |
| RETURNED CHECK RESOLUTION INQUIRY | 04500000 |
| RETURNED CHECK RESOLUTION NON CITATION UPDATE | 04520000 |
| RETURNED CHECK RESOLUTION CITATION UPDATE | 04540000 |
| CCP - CMVI CITATION (INQ BY CITA) | 04570000 |
| RETURNED CHECK RESOLUTION CITATION UPDATE | 04590000 |
| CCP - CMVI CITATION (INQ BY LICENSE OR NOWNM NAME) | 04610000 |
| CCP - CMVI CITATION (UPDATE) | 04630000 |
| DATE - COMPARE PC DATE TO EIBDATE | 04650000 |
| DI01 - MULTIPLE PLATE INQUIRY SECTION V | 04670000 |
| DRVP - VANITY PLATE ORDER SECTION V | 04690000 |
* DR01 - PLATE ADD AND AMEND FOR SECTION V 04710000
  PCTEH DR01, PGM=(UHR1803P), RMV=DRPL, SERVICE=1 04720000
* DR02 - ADDRESS ADD AND AMEND FOR SECTION V 04730000
  PCTEH DR02, PGM=(UHR1813P), RMV=DRAD, SERVICE=1 04740000
* 04750000
* EXAM DISTANCE CHECK, PROD LOOKUP 04760000
  PCTEH EBDC, PGM=(UHB2040P), RMV=EBDC 04770000
* EXAM DISTANCE CHECK, TEST LOOKUP 04780000
  PCTEH EBDD, PGM=(UHB2040P), FLAGS=00EE000000000000, RMV=EBDC 04790000
* EXAM DISTANCE CHECK, TEST LOOKUP 04800000
  PCTEH EBDT, PGM=(UHB2040P), FLAGS=C900000000000000, RMV=EBDC 04810000
* EXAM DISTANCE CHECK, TEST LOOKUP 04820000
  PCTEH EBDU, PGM=(UHB2040P), FLAGS=C9EE000000000000, RMV=EBDC 04830000
* 04840000
* EXAM LOCATION INQUIRY 04850000
  PCTEH EBLI, PGM=(UHB2020P), RMV=EBLM 04860000
* EXAM LOCATION MAINTENANCE 04870000
  PCTEH EBLM, PGM=(UHB2020P), FLAGS=E4000000000000000, RMV=EBLM 04880000
* EXAM OFFICE INQUIRY 04890000
  PCTEH EBOI, PGM=(UHB2060P), RMV=EBOM 04900000
* EXAM OFFICE MAINTENANCE 04910000
  PCTEH EBO, PGM=(UHB2060P), RMV=EBOM 04920000
* EXAM ZIP INQUIRY 04930000
  PCTEH EBZI, PGM=(UHB2000P), RMV=EBZM 04940000
* EXAM ZIP MAINTENANCE 04950000
  PCTEH EBZM, PGM=(UHB2000P), FLAGS=E4000000000000000, RMV=EBZM 04960000
* 04970000
* EXAM BOOKING SCHEDULE DATASET FETCH AND UPDATE (PRE-CDL) 04980000
  PCTEH EB01, PGM=(UHB0010P), RMV=ED, SERVICE=8 04990000
* EXAM HISTORY DATASET FETCH AND UPDATE 05000000
  PCTEH EB02, PGM=(UHB0020P), RMV=EB, SERVICE=8 05010000
* 05020000
* EXAM BOOKING SCHEDULE DATASET FETCH AND UPDATE (CDL IMPLEMENTATION) 05030000
  PCTEH EB03, PGM=(UHB0030P), RMV=ED, SERVICE=11 05040000
* EXAM BOOKING GRADE MULTIPLE PENDING EXAMS FOR IMAGING 05050000
  PCTEH EB05, PGM=(UHB0050P), RMV=EB, SERVICE=39 05060000
* EXAM BOOKING GET EXAM HISTORY FOR CUSTOMER 05070000
  PCTEH EB06, PGM=(UHB0060P), RMV=EGCO 05080000
* EXAM BOOKING EXAM BLOCK APPOINTMENT UPDATE 05090000
  PCTEH EB40, PGM=(UHB0040P), RMV=ED, SERVICE=10 05100000
* FCP - FOREIGN CASH POSTING - VERSION 1 05110000
  PCTEH FC01, PGM=(UHC1023P), RMV=FCP 05120000
* 05130000
* HOST VSAM IO FUNCTIONS 05140000
* EVR SOLICITED AND UNSOLICITED FILES 05150000
  PCTEH H100, PGM=(UHZ0060P, UHL0200P, UHZ0061P) 05160000
  PCTEH H101, PGM=(UHZ0060P, UHL0200P, UHZ0061P) 05170000
* TABLE LOAD FUNCTIONS: 05180000
  PCTEH HTB0, PGM=(UHZ0004P, UHZ0006P) 05190000
  PCTEH HTB1, PGM=(UHZ0004P) 05200000
  PCTEH HTB2, PGM=(UHZ0004P, MRMWAFUM) 05210000
  PCTEH HTB3, PGM=(UHZ0004P, UHZ0007P) 05220000
  PCTEH HTB4, PGM=(UHZ0004P, UHZ0102P) 05230000
  PCTEH HTB5, PGM=(UHZ0004P, UHZ0103P) 05240000
  PCTEH HTB6, PGM=(UHZ0004P, UHZ0104P) 05250000
  PCTEH HTB7, PGM=(UHZ0004P, UHZ0105P) 05260000
  PCTEH HTB8, PGM=(UHZ0004P, UHZ0102P) 05270000
* IMAGE SERVER HOST FUNCTIONS:  

PCTEH IHA0, PGM=(UHL1000P), RMV=LZI1, SERVICE=13,  
FLAGS=C1000000000000000  
X05290000  

PCTEH IHB0, PGM=(UHL1020P), RMV=LZI1, SERVICE=14,  
FLAGS=C2000000000000000  
X05310000  

PCTEH IHC0, PGM=(UHL1000P), RMV=LZI1, SERVICE=13,  
FLAGS=C3000000000000000  
X05330000  

PCTEH IHD0, PGM=(UHL1000P), RMV=LZI1, SERVICE=13,  
FLAGS=C4000000000000000  
X05350000  

PCTEH IHE0, PGM=(UHL1000P), RMV=LZI2, SERVICE=13,  
FLAGS=C5000000000000000  
X05370000  

PCTEH IHG0, PGM=(UHL1040P), RMV=LZI2, SERVICE=13,  
FLAGS=C7000000000000000  
X05400000  

* IDENTICAL TO IHG0 EXCEPT SECURITY COMES FROM UDUP  
PCTEH IHG1, PGM=(UHL1040P), RMV=UDUP, SERVICE=13,  
FLAGS=C7000000000000000  
X05420000  

* IDENTICAL TO IHG0, BUT FORCES A DELAY OF ZERO.  
* THIS IS USED BY BULK RESUBMISSION, TRIGGERED BY FLAGS+1=01  
PCTEH IHG2, PGM=(UHL1040P), RMV=LZI2, SERVICE=31,  
FLAGS=C7010000000000000  
X05460000  

PCTEH IHH0, PGM=(UHL1000P), RMV=LZI2, SERVICE=13,  
FLAGS=C8000000000000000  
X05480000  

PCTEH IHH0, PGM=(UHL1010P), RMV=LZI2, SERVICE=13,  
FLAGS=C9000000000000000  
X05500000  

PCTEH IHI0, PGM=(UHL1040P), RMV=LZI2, SERVICE=13,  
FLAGS=D1000000000000000  
X05530000  

PCTEH IHO0, PGM=(UHL1000P), RMV=LZI3, SERVICE=13,  
FLAGS=D2000000000000000  
X05540000  

PCTEH IHL0, PGM=(UHL1020P), RMV=LZI1, SERVICE=13,  
FLAGS=D3000000000000000  
X05560000  

PCTEH IHH0, PGM=(UHL1000P), RMV=LZI1, SERVICE=13,  
FLAGS=D4000000000000000  
X05580000  

PCTEH IHN0, PGM=(UHL1000P), RMV=LZI3, SERVICE=13,  
FLAGS=D5000000000000000  
X05600000  

PCTEH IHP0, PGM=(UHL1000P), RMV=LZI3, SERVICE=13,  
FLAGS=D7000000000000000  
X05620000  

PCTEH IHQ0, PGM=(UHL1000P), RMV=LZI2, SERVICE=13,  
FLAGS=D8000000000000000  
X05640000  

PCTEH IHT0, PGM=(UHL1000P), RMV=LZI2, SERVICE=13,  
FLAGS=D8000000000000000  
X05660000  

* ILA2 - UPDATE TRAINING INFORMATION BY RMV  
PCTEH ILA2, PGM=(UHE0102P), RMV=ILRU, SERVICE=40  
X05720000  

* IL01 - INSPECTOR TRAINING AUTHORIZATION  
PCTEH IL01, PGM=(UHE0101P), RMV=ILVI, SERVICE=40  
X05730000  

* IL02 - UPDATE TRAINING INFORMATION BY VENDOR  
PCTEH IL02, PGM=(UHE0102P), RMV=ILVU, SERVICE=40  
X05740000  

* IL03 - ADD APPLICANT TO DATA BASE  
PCTEH IL03, PGM=(UHE0103P), RMV=ILRU, SERVICE=40  
X05750000  

* IL04 - AUTHORIZE INSPECTOR  
PCTEH IL04, PGM=(UHE0104P), RMV=ILVI, SERVICE=40  
X05760000  

* IL05 - RETRIEVE INSPECTOR/APPLICANT INFORMATION  
PCTEH IL05, PGM=(UHE0105P), RMV=ILRI  
X05770000  

* IL06 - UPDATE INSPECTOR/APPLICANT INFORMATION  
PCTEH IL06, PGM=(UHE0106P), RMV=ILRU, SERVICE=40  
X05780000  

* IL07 - DELETE INSPECTOR LICENSE  
PCTEH IL07, PGM=(UHE0107P), RMV=ILDI, SERVICE=40  
X05790000  

* IL08 - DELETE APPLICANT  
X05800000  

X05810000  

X05820000  

X05830000  

X05840000  

X05850000  

X05860000  

X05870000  

X05880000
PCTEH IL08, PGM=(UHE0107P), RMV=ILRU, SERVICE=40 05890000
* IL09 - ISSUE INSPECTOR LICENSE
  PCTEH IL09, PGM=(UHE0109P), RMV=ILRU, SERVICE=40 05900000
* IL10 - PRINT INSPECTOR LICENSE
  PCTEH IL10, PGM=(UHE0110P), RMV=ILRU, SERVICE=40 05910000
* IL13 - RETRIEVE INSPECTOR LICENSE HISTORY
  PCTEH IL13, PGM=(UHE0113P), RMV=ILRI, SERVICE=40 05930000
* IL14 - RENEW INSPECTOR LICENSE
  PCTEH IL14, PGM=(UHE0114P), RMV=ILRU, SERVICE=40 05940000
* IMU - INSPECTION HISTORY UPDATE INPUT
  PCTEH IM01, PGM=(UHI1353P), RMV=IMU, SERVICE=3 05950000
* IMU - INSPECTION HISTORY UPDATE OUTPUT
  PCTEH IM02, PGM=(UHI1354P), RMV=IMU 05960000
* IMU - INSPECTION UPDATE
  PCTEH IM02, PGM=(UHI1353P), RMV=IMU 05970000
* IMUQ - INSPECTION DRIVER CONTROL
  PCTEH IM02, PGM=(UHI1353P), RMV=IMU 05980000
* IMUQ - INSPECTION HISTORY INQUIRY
  PCTEH IM04, PGM=(UHI1343P), RMV=IMU 05990000
*************************************************************** 06000000
* IMAGING PRINTING OF FREE-FORM DOCUMENTS
*************************************************************** 06010000
* GET DOCUMENT TEMPLATE
  PCTEH IP01, PGM=(UHU3100P) 06020000
* GET VARIABLE DATA
  PCTEH IP02, PGM=(UHU3200P) 06030000
* GET DOCUMENT DEFAULT PRINTER
  PCTEH IP03, PGM=(UHU3300P) 06040000
*************************************************************** 06050000
* IMAGING SECURITY:
  PCTEH ISEC, PGM=(UHZ0030P, UHZ0032P, UHZ0031P), RMV=LI 06060000
* MAB PRINT MODULE SELECTOR
  PCTEH IW01, PGM=(IWP1000P), RMV=LI 06070000
* IDENTICAL TO IHE0 (USED FOR AUTOMATIC STORAGE AUTHORIZATION REQUEST)
* USED TO RECOVER LOST IMAGES
  PCTEH IXE0, PGM=(UHL1000P), RMV=LZI2, SERVICE=13, X06080000
  FLAGS=C500000000000000 06090000
* REPORT IMAGE NOT TRANSFERRED TO HOST FROM IMAGE SERVER
  PCTEH IXF0, PGM=(UHL1130P, UHL0200P), RMV=LZI1 06100000
* INTERNAL FUNCTION (USED TO DELETE TRACKING RECORDS)
  PCTEH IXZZ, PGM=(UHL1000P), RMV=LZI2, SERVICE=13, X06110000
  FLAGS=000000000000000 06120000
* LICENSE ADDRESS CHANGE HIST BY SURR
  PCTEH LAC1, PGM=UHL3320P, RMV=LI 06130000
  PCTEH LAI1, PGM=(UHL0070P, UHL3300P), RMV=LI, DUPKEY=L104 06140000
* LICENSE ADDRESS INFORMATION BY SURR
  PCTEH LAI2, PGM=UHL3300P, RMV=LI 06150000
* GENERALIZED FETCHER OF BREFS:
  PCTEH LB10, PGM=(UHL0420P), RMV=LI 06160000
* GENERALIZED FETCHER OF BREF BATCH NUMBERS:
  PCTEH LB11, PGM=(UHL1140P), RMV=LI, SERVICE=30 06170000
* DUPLICATE SSN SCROLL
  PCTEH LD04, PGM=(UHL0500P), RMV=LP 06180000
* DUPLICATE OOS SCROLL
  PCTEH LD08, PGM=(UHL0520P), RMV=LP 06190000
* LICENSE HISTORY LOOK-UP BY LICENSE #
  PCTEH LH10, PGM=(UHL0070P, UHL0060P), X06200000
  FLAGS=000000000000000 06210000
* LICENSE ADDRESS CHANGE HIST BY SURR
  PCTEH LAC1, PGM=UHL3320P, RMV=LI 06220000
  PCTEH LAI1, PGM=(UHL0070P, UHL3300P), RMV=LI, DUPKEY=L104 06230000
* LICENSE ADDRESS INFORMATION BY SURR
  PCTEH LAI2, PGM=UHL3300P, RMV=LI 06240000
* GENERALIZED FETCHER OF BREFS:
  PCTEH LB10, PGM=(UHL0420P), RMV=LI 06250000
* GENERALIZED FETCHER OF BREF BATCH NUMBERS:
  PCTEH LB11, PGM=(UHL1140P), RMV=LI, SERVICE=30 06260000
* DUPLICATE SSN SCROLL
  PCTEH LD04, PGM=(UHL0500P), RMV=LP 06270000
* DUPLICATE OOS SCROLL
  PCTEH LD08, PGM=(UHL0520P), RMV=LP 06280000
* LICENSE HISTORY LOOK-UP BY LICENSE #
  PCTEH LH10, PGM=(UHL0070P, UHL0060P), X06290000
  FLAGS=000000000000000 06300000
* REPORT IMAGE NOT TRANSFERRED TO HOST FROM IMAGE SERVER
  PCTEH IXF0, PGM=(UHL1130P, UHL0200P), RMV=LZI1 06310000
* INTERNAL FUNCTION (USED TO DELETE TRACKING RECORDS)
  PCTEH IXZZ, PGM=(UHL1000P), RMV=LZI2, SERVICE=13, X06320000
  FLAGS=000000000000000 06330000
* LICENSE ADDRESS CHANGE HIST BY SURR
  PCTEH LAC1, PGM=UHL3320P, RMV=LI 06340000
  PCTEH LAI1, PGM=(UHL0070P, UHL3300P), RMV=LI, DUPKEY=L104 06350000
* LICENSE ADDRESS INFORMATION BY SURR
  PCTEH LAI2, PGM=UHL3300P, RMV=LI 06360000
  PCTEH LAI1, PGM=(UHL0070P, UHL3300P), RMV=LI, DUPKEY=L104 06370000
* LICENSE ADDRESS INFORMATION BY SURR
  PCTEH LAI2, PGM=UHL3300P, RMV=LI 06380000
* GENERALIZED FETCHER OF BREFS:
  PCTEH LB10, PGM=(UHL0420P), RMV=LI 06390000
* GENERALIZED FETCHER OF BREF BATCH NUMBERS:
  PCTEH LB11, PGM=(UHL1140P), RMV=LI, SERVICE=30 06400000
* DUPLICATE SSN SCROLL
  PCTEH LD04, PGM=(UHL0500P), RMV=LP 06410000
* DUPLICATE OOS SCROLL
  PCTEH LD08, PGM=(UHL0520P), RMV=LP 06420000
* LICENSE HISTORY LOOK-UP BY LICENSE #
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMV=LI, DUPKEY=LI04</td>
<td>License History Look-Up by OOS #</td>
<td>06540000</td>
</tr>
<tr>
<td>PCTEH LH20, PGM=(UHL0080P, UHL0060P), RMV=LI, DUPKEY=LD08</td>
<td>License History Look-Up by Surrogate</td>
<td>06580000</td>
</tr>
<tr>
<td>PCTEH LH30, PGM=(UHL0060P), RMV=LI</td>
<td>License History Commercial Extension -- Surrogate Only</td>
<td>06590000</td>
</tr>
<tr>
<td>PCTEH LH40, PGM=(UHL0095P), RMV=LI</td>
<td>License Unisite Group Lookup:</td>
<td>06600000</td>
</tr>
<tr>
<td>PCTEH LIH, PGM=(UHL0261P), RMV=LI</td>
<td>License Image History Lookup:</td>
<td>06610000</td>
</tr>
<tr>
<td>PCTEH LI01, PGM=(UHL0010P), RMV=LI, DUPKEY=LI04</td>
<td>License Inquiry</td>
<td>06620000</td>
</tr>
<tr>
<td>PCTEH LI02, PGM=(UHL0030P), RMV=LI</td>
<td>License Person Scroll</td>
<td>06630000</td>
</tr>
<tr>
<td>PCTEH LI03, PGM=(UHL0040P), RMV=LP, SERVICE=17</td>
<td>License Number Scroll</td>
<td>06640000</td>
</tr>
<tr>
<td>PCTEH LI04, PGM=(UHL0020P), RMV=LP</td>
<td>License History</td>
<td>06650000</td>
</tr>
<tr>
<td>PCTEH LI06, PGM=(UHL0060P), RMV=LI</td>
<td>License Inquiry (By Lic. Number) (New)</td>
<td>06660000</td>
</tr>
<tr>
<td>PCTEH LI07, PGM=(UHL0070P, UHL0100P, UHL0110P), SERVICE=6, RMV=LI, DUPKEY=LI04, FLAGS=0003000000000000</td>
<td>License Inquiry (By OOS Number) (New)</td>
<td>06670000</td>
</tr>
<tr>
<td>PCTEH LI08, PGM=(UHL0080P, UHL0100P, UHL0110P), SERVICE=6, RMV=LI, DUPKEY=LD08, FLAGS=0003000000000000</td>
<td>License Inquiry (By OOS Number) (New)</td>
<td>06680000</td>
</tr>
<tr>
<td>PCTEH LI09, PGM=(UHL0090P, UHL0100P, UHL0110P), SERVICE=6, RMV=LI, DUPKEY=LI04, FLAGS=0003000000000000</td>
<td>License Inquiry (By Surrogate) (New)</td>
<td>06690000</td>
</tr>
<tr>
<td>PCTEH LI11, PGM=(UHL0110P), RMV=LI, SERVICE=6</td>
<td>Endorsement Inquiry by Surrogate</td>
<td>06700000</td>
</tr>
<tr>
<td>PCTEH LI12, PGM=(UHL0120P), RMV=LI</td>
<td>CDL Driver Type Host Function</td>
<td>06710000</td>
</tr>
<tr>
<td>PCTEH LI14, PGM=(UHL0140P), RMV=LI</td>
<td>S-Number Generator Host Function</td>
<td>06720000</td>
</tr>
<tr>
<td>PCTEH LI15, PGM=(UHL0150P), RMV=LC</td>
<td>Get Control Record</td>
<td>06730000</td>
</tr>
<tr>
<td>PCTEH LI17, PGM=(UHL0540P), RMV=LC, SERVICE=4</td>
<td>Get License Security</td>
<td>06740000</td>
</tr>
<tr>
<td>PCTEH LI18, PGM=(UHL1100P), RMV=LC</td>
<td>NDR Inquiry Host Function</td>
<td>06750000</td>
</tr>
<tr>
<td>PCTEH LI19, PGM=(UHL1080P), RMV=LC, SERVICE=26</td>
<td>Customer Add Host Function</td>
<td>06760000</td>
</tr>
<tr>
<td>PCTEH LI20, PGM=(UHL1060P), RMV=LC, SERVICE=15</td>
<td>Create Permit Application Host Function</td>
<td>06770000</td>
</tr>
<tr>
<td>PCTEH LI21, PGM=(UHL0201P), SERVICE=19, RMV=LC</td>
<td>Issue Permit Host Function</td>
<td>06780000</td>
</tr>
<tr>
<td>PCTEH LI22, PGM=(UHL0202P), SERVICE=19, RMV=LC</td>
<td>Issue License Host Function</td>
<td>06790000</td>
</tr>
<tr>
<td>PCTEH LI23, PGM=(UHL0203P), SERVICE=19</td>
<td>X07100000</td>
<td></td>
</tr>
</tbody>
</table>
RMV=LIC4
* ISSUE MASS & LIQUOR ID HOST FUNCTION
  PCTEH LI24,PGM=(UHL0204P),SERVICE=19,
  RMV=LM
  X07130000
  07140000
* UPDATE LICENSE EXPIRATION DATE FUNCTION
  PCTEH LI25,PGM=(UHL0205P),SERVICE=48,
  RMV=LIC4
  X07160000
  07170000
* DELETE MASS & LIQUOR ID HOST FUNCTION
  PCTEH LI26,PGM=(UHL0206P),RMV=LM
  X07190000
* DELETE PERMIT RECORD HOST FUNCTION
  PCTEH LI27,PGM=(UHL0207P),SERVICE=19,
  RMV=LIC2
  X07210000
  07220000
* GET FEE TABLE RECORD HOST FUNCTION
  PCTEH LI28,PGM=(UHL0208P)
  07230000
  07240000
* STORE AND MODIFY MMVR-PSSN
  PCTEH LI29,PGM=(UHL0209P),RMV=LC
  07260000
* STORE MMVR-UNOP
  PCTEH LI30,PGM=(UHL0210P),RMV=LC
  07280000
* MODIFY MMVR-UNOP
  PCTEH LI31,PGM=(UHL0211P),RMV=LC
  07300000
* LICENSE NUMBER CHANGE
  PCTEH LI32,PGM=(UHL0212P),RMV=LC
  07320000
* CASH POSTING
  PCTEH LI33,PGM=(UHL0213P),SERVICE=19
  07340000
  07360000
* VERIFY FEE
  PCTEH LI34,PGM=(UHL0214P),SERVICE=19
  07350000
* DELETE LICENSE
  PCTEH LI35,PGM=(UHL0215P),SERVICE=19,RMV=LIC4
  07380000
  07390000
* EXAM OVERRIDE
  PCTEH LI36,PGM=(UHL0216P),SERVICE=26,RMV=LIC2
  07400000
* REMOVE EXAM OVERRIDE
  PCTEH LI37,PGM=(UHL0217P),SERVICE=19,RMV=LIC2
  07420000
* CREDIT CARD AUTHORIZATION
  PCTEH LI38,PGM=(UHL0218P),SERVICE=20,RMV=LIC4
  07440000
* RXX REINSTATE LICENSE
  PCTEH LI39,PGM=(UHL0219P),SERVICE=21,RMV=LIC4
  07450000
  07460000
* RXW REINSTATE LICENSE
  PCTEH LI40,PGM=(UHL0245P),SERVICE=21,RMV=LIC4
  07480000
* RXP COLLECT REINSTATEMENT FEE AND APPLY FOR EXAM
  PCTEH LI41,PGM=(UHL0221P),SERVICE=19,RMV=LR
  07500000
* BUILD DESCRIPTIVE TABLE
  PCTEH LI42,PGM=(UHL0222P),SERVICE=13
  07520000
* FORCED LICENSE CLASS DOWNGRADE
  PCTEH LI43,PGM=(UHL0223P),SERVICE=21,RMV=LIC4
  07540000
* RXW ISSUE PERMIT
  PCTEH LI44,PGM=(UHL0224P),SERVICE=19,RMV=LIC4
  07550000
  07560000
* RETRIEVE LICENSE DATA
  PCTEH LI45,PGM=(UHL0225P),SERVICE=16
  07580000
  07590000
* UPDATE ENDORSEMENT DATA
  PCTEH LI46,PGM=(UHL0226P),SERVICE=16,RMV=LIC1
  07600000
  07610000
* SNAP PRINT OF LICENSE/ENDORSEMENTS/RESTRICTION
  PCTEH LI47,PGM=(UHL0470P),RMV=LI,SERVICE=12
  07620000
* UPDATE RESTRICTION DATA
  PCTEH LI48,PGM=(UHL0228P,UHL0400P),SERVICE=21,RMV=LR
  07630000
  07640000
* I OR K RESTRICTION ASSIGNMENT
  PCTEH LI49,PGM=(UHL0229P),SERVICE=51,RMV=LR
  07650000
  07660000
* STAND ALONE CASH POSTING FOR APWS
  PCTEH LI49,PGM=(UHL0229P),SERVICE=51,RMV=LR
  07670000

Appendix C – Host Process Control Table
PCTEH LI5A,PGM=(UHL0510P),SERVICE=19 07680000
* STAND ALONE CASH POSTING
PCTEH LI50,PGM=(UHL0230P),SERVICE=19 07700000
* NDR INQUIRY HOST FUNCTION
PCTEH LI51,PGM=(UHL0235P,UHL0231P),SERVICE=19, X07720000
FLAGS=C440000000000000,RMV=LX 07730000
* CHECK FOR BAD CHECK INTERFACE FUNCTION
PCTEH LI52,PGM=(UHL0232P),SERVICE=23 07750000
* UPDATE PERSONAL RECORD WITH GENERATED LICENSE NUMBER
PCTEH LI53,PGM=(UHL0233P),RMV=LC,SERVICE=15 07770000
* OBTAIN OUT OF STATE DATA
PCTEH LI54,PGM=(UHL0234P),RMV=LIC1,SERVICE=16 07790000
* ISSUE LICENSE HOST FUNCTION OOS CONVERSION
PCTEH LI55,PGM=(UHL0203P),SERVICE=19, X07810000
RMV=LIC4 07820000
* CREATE LICENSE APPLICATION HOST FUNCTION
*** PCTEH LI56,PGM=(UHL0236P),SERVICE=19, X 07840000
PCTEH LI56,PGM=(UHL0236P),SERVICE=33, X07850000
RMV=LIC3 07860000
* ISSUE PERMIT HOST FUNCTION (CONVERSION)
PCTEH LI57,PGM=(UHL0202P),SERVICE=19, X07880000
RMV=LIC2 07890000
* CASH POSTING FOR PROCESS ENDORSEMENT
PCTEH LI58,PGM=(UHL0238P),SERVICE=19 07910000
* RETRIEVE DESCRIPTION FROM MMVR-VTAB
PCTEH LI59,PGM=(UHL0239P),SERVICE=19,RMV=LI 07930000
* PLACARD RECORD CANCEL OR PURGE
PCTEH LI60,PGM=(UHL0680P),SERVICE=28,RMV=PLCP 07950000
* OUT OF STATE LICENSE CONVERSION
PCTEH LI61,PGM=(UHL0680P),SERVICE=28,RMV=PLCD 07970000
* UPDATE UNPAID WRITTEN EXAMS
PCTEH LI62,PGM=(UHL0690P),SERVICE=28,RMV=PLCD 07990000
* RETRIEVE DESCRIPTION FROM MMVR-ORES
PCTEH LI63,PGM=(UHL0243P),SERVICE=19,RMV=LI 08010000
* CALCULATE REINSTATEMENT FEES
PCTEH LI64,PGM=(UHL0244P),SERVICE=19,RMV=LI 08030000
* RXW REINSTATE LICENSE RENEW
PCTEH LI65,PGM=(UHL0245P),SERVICE=21,RMV=LIC4 08050000
* RXP COLLECT FEE AND PRINT LICENSE
PCTEH LI66,PGM=(UHL0221P),SERVICE=19,RMV=LIC1 08070000
* P-NUMBER GENERATOR HOST FUNCTION
PCTEH LI67,PGM=(UHL0670P),RMV=LC 08090000
* PLACARD RECORD UPDATE
PCTEH LI68,PGM=(UHL0680P),SERVICE=28,RMV=PLCD 08110000
* PLACARD RECORD LOOKUP
PCTEH LI69,PGM=(UHL0690P) 08130000
* NOSSN OVERRIDE / REMOVE OVERRIDE
PCTEH LI70,PGM=(UHL0700P),SERVICE=19 08140000
* BUILD DEPT, DEPT NAME, DEPT EXT TABLE FOR ACTIVITY HOLD
PCTEH LI71,PGM=(UHL0710P),SERVICE=49 08160000
* AUTOMATIC ADJUST MANUFACTURE
PCTEH LI72,PGM=(UHL0720P),SERVICE=19 08180000
* CUSTOMER ACTIVITY FUNCTION
PCTEH LI73,PGM=(UHL0730P),SERVICE=30,RMV=LIC1 08210000
* RETRIEVE MMVR-PERS-EXT (ACTIVITY HOLD)
PCTEH LI74,PGM=(UHL0740P),SERVICE=49,RMV=LIC4 08230000
* ISSUE LICENSE WITHOUT "O" RESTRICTION
08240000
PCTEH LI75, PGM=(UHL0750P), SERVICE=19, RMV=LIC4 08250000
* MAB SURRENDER UPDATE FUNCTION
  PCTEH LI76, PGM=(UHL0760P), SERVICE=27, RMV=MSRU 08270000
* 08280000
* MAB SURRENDER INQUIRY FUNCTION
  PCTEH LI77, PGM=(UHL0770P), RMV=MSRI 08300000
* 08310000
* HANDICAPPED PERS-EXT BUILD
  PCTEH LI78, PGM=(UHL0078P), SERVICE=13, RMV=LZI1, X08330000
  FLAGS=C3000000000000000 08340000
* RETRIEVE OFFC-NAME FROM MMVR-OFFC USING OFFC-NUMB
  PCTEH LI79, PGM=(UHL0790P) 08360000
* SSN VERIFICATION
* GT0400 PCTEH LI80, PGM=(UHL0081P), SERVICE=13 08380000
* LI81 - UPDATE LICENSING PAYMENT INFORMATION
  PCTEH LI81, PGM=(UHL0810P), SERVICE=38, RMV=LIC4 08400000
* LI82 - RETRIEVE RESTRICTION RECORDS
  PCTEH LI82, PGM=(UHL0180P) 08420000
* LI83 - ADD / UPDATE RESTRICTION RECORDS
  PCTEH LI83, PGM=(UHL0190P) 08440000
* LICENSE PERSON UPDATE (NEW)
  PCTEH LI85, PGM=(UHL0085P), RMV=LC, SERVICE=51 08460000
* SSN VERIFICATION AND RESOLUTION
  PCTEH LI87, PGM=(UHL0870P), RMV=LSSN, SERVICE=4 08480000
* SSN VERIFICATION REQUEST
  PCTEH LI88, PGM=(UHL0880P), SERVICE=13, RMV=LI 08500000
  08510000
* VALIDATE DEMOGRAPHICS FOR L1 SCREEN
  PCTEH LI89, PGM=(UHL0890P) 08530000
  08540000
* LI90 - MAINTAIN THE PENDING ISSUE TRANSACTION FILE
  PCTEH LI90, PGM=(UHL4000P), SERVICE=8 08560000
  08570000
* LI91 - RETRIEVE THE PENDING ISSUE TRANSACTIONS
  PCTEH LI91, PGM=(UHL4500P), SERVICE=45 08590000
  08600000
* LI92 - READ THE PAYMENT LOG FILE
  PCTEH LI92, PGM=(UHL0920P) 08620000
  08630000
* LI93 - UPDATE THE PAYMENT LOG FILE
  PCTEH LI93, PGM=(UHL0930P) 08650000
  08660000
* LI94 - ISSUE PERMIT HOST FUNCTION (ISSUE DATE)
  PCTEH LI94, PGM=(UHL0202P), SERVICE=19, X08680000
  RMV=LIC2, FLAGS=C9000000000000000 08690000
* LI95 - RETRIEVE PAYMENT LOG
  PCTEH LI95, PGM=(UHL0940P) 08710000
  08720000
* LI96 - GENERATE EXPIRATION DATE (ONLY PERMIT FOR NOW)
  PCTEH LI96, PGM=(UHL0960P) 08740000
  08750000
* LI97 - LICENSE ISSUE FOR MOTORCYCLE COURSE
  PCTEH LI97, PGM=(UHL0970P), SERVICE=47 08770000
  08780000
* LI98 - DETERMINE ACTIVITY HOLD STATUS
  PCTEH LI98, PGM=(UHL0980P) 08800000
  08810000
* MANUFACTURE HISTORY INQUIRE 08820000
  PCTEH LM01,PGM=(UHL3200P),SERVICE=4,RMV=LI 08830000
* LOTTERY INQUIRY FUNCTION 08840000
  PCTEH LOT1,PGM=(UHN0200P),SERVICE=9,RMV=RI 08850000
* SOCIAL SECURITY NUMBER DELETE FUNCTION 08860000
  PCTEH LSSD,PGM=(UHL1200P,UHL0400P),RMV=LSSD 08870000
* 08880000
* MOTOR VOTER PERS-EXT BUILD 08890000
  PCTEH LV20,PGM=(UHV0020P),SERVICE=13,RMV=LZI1, X08900000
  FLAGS=C3000000000000000 08910000
* MOTOR VOTER REGISTRATION HISTORY 08920000
  PCTEH LV30,PGM=(UHV0030P),RMV=LIMV 08930000
* MOTOR VOTER NAME AND ADDRESS INFORMATION 08940000
  PCTEH LV31,PGM=(UHV0031P),RMV=LIMV 08950000
* MOTOR VOTER REGISTER TO VOTE 08960000
  PCTEH LV32,PGM=(UHV0032P),SERVICE=13,RMV=LZI1, X08970000
  FLAGS=C3000000000000000 08980000
* MOTOR VOTER DELETE VOTER REGISTRATION 08990000
  PCTEH LV33,PGM=(UHV0033P),RMV=LIMV 09000000
* MOTOR VOTER RE-SEND VOTER REGISTRATION INFORMATION 09010000
  PCTEH LV34,PGM=(UHV0034P),RMV=LIMV 09020000
* MOTOR VOTER HISTORY LOOKUP 09030000
  PCTEH LV40,PGM=(UHV0040P),RMV=LHMOV 09040000
* 09050000
* PEEL IMAGE OFF TO IMAGE UNLOAD DATASET 09060000
  PCTEH LV50,PGM=(UHV0050P),RMV=LZI1, X09070000
  FLAGS=C3000000000000000,SERVICE=13 09080000
* 09090000
* ENTRIES UNIQUE TO LX OR THE GHOST: 09100000
* GENERALIZED TRANSFER-FILE INQUIRE_STATUS: 09110000
  PCTEH LXA0,PGM=(UHL0440P),SERVICE=4,RMV=LX 09120000
* GENERALIZED TRANSFER-FILE INQUIRE_DETAIL: 09130000
  PCTEH LXBO,PGM=(UHL0460P),SERVICE=4,RMV=LX 09140000
* GENERALIZED TRANSFER-FILE EMULSIFY NO BREF: 09150000
  PCTEH LXCO,PGM=(UHL0480P),SERVICE=4,RMV=LX, X09160000
  FLAGS=8000000000000000 09170000
* GENERALIZED TRANSFER-FILE EMULSIFY WITH BREF: 09180000
  PCTEH LXCI,PGM=(UHL0480P,UHL0410P),SERVICE=4,RMV=LX 09190000
* GENERALIZED INQUIRE_CDL: 09200000
  PCTEH LXDO,PGM=(UHL0380P,UHLNDR1P,UHL0410P),SERVICE=4,RMV=LX 09210000
* CDLIS OVERRIDE BREF: (1ST 2 FLAG BYTES=SUBFUNCTION SECURITY) 09220000
  PCTEH LXE0,PGM=(UHL0490P,UHL0410P),SERVICE=4,RMV=LYR, X09230000
  FLAGS=D6400000000000000 09240000
* 09250000
* GENERALIZED CDL AUTO APPLY: 09260000
  PCTEH LXF0,PGM=(UHL0430P,UHL0450P,UHL0410P),SERVICE=4,RMV=LX 09270000
* GENERALIZED CDL AUTO APPLY: 09280000
  PCTEH LXF1,PGM=(UHL0430P,UHL0450P,UHL0410P),SERVICE=4,RMV=LX, X09290000
  FLAGS=8000000000000000 09300000
* GENERALIZED CDL MESSAGE CONTROL RECORD REAL TIME SEND CLOCK UPDATE 09310000
  PCTEH LXGO,PGM=(UHL0450P),SERVICE=4,RMV=LX 09320000
* FORMAT PDPS MESSAGE REQUESTS 09330000
  PCTEH LXPO,PGM=(UHL0600P),SERVICE=26, X09340000
  RMV=LX 09350000
* PDPS PERS-EXT TYPE 6 LOOKUP 09360000
  PCTEH LXPl,PGM=(UHL0610P),RMV=LI 09370000
* PDPS BREF WRITE, PERS-EXT UPDATE 09380000
PCTEH LXP2, PGM=(UHL0630P, UHL0620P), SERVICE=4, RMV=LX

* LX UPDATE_DETAIL:
  PCTEH LX10, PGM=(UHL0260P, UHL0340P, UHL0400P), SERVICE=4, RMV=LX

* DV0196
  PCTEH LX11, PGM=(UHL0260P, UHL0340P, UHL0400P), SERVICE=4, RMV=LX,

  FLAGS=C4400000000000000

X09440000

* DV0196 END

* LX INQUIRE_DETAIL:
  PCTEH LX20, PGM=(UHL0240P), SERVICE=4, RMV=LX

* LX INQUIRE_STATUS:
  PCTEH LX40, PGM=(UHL0280P), SERVICE=4, RMV=LX

* LX INQUIRE_NDR:
  PCTEH LX50, PGM=(UHL0300P, UHLNDR1P, UHL0400P), SERVICE=4, RMV=LX

* LXGHOST INQUIRE_NDR
  PCTEH LX51, PGM=(UHL0300P, UHLNDR1P, UHL0400P), SERVICE=4, RMV=LX,

  FLAGS=8000000000000000

X09540000

* LX EMULSIFY:
  PCTEH LX60, PGM=(UHL0320P, UHL0400P), SERVICE=4, RMV=LX

* LXGHOST EMULSIFY:
  PCTEH LX61, PGM=(UHL0320P, UHL0400P), SERVICE=4, RMV=LX,

  FLAGS=8000000000000000

X09590000

* LX-WRITE OVERRIDE BREF: (1ST 2 FLAG BYTES=SUBFUNCTION SECURITY)
  PCTEH LX70, PGM=(UHL0260P, UHL0400P), SERVICE=4, RMV=LX,

  FLAGS=C4400000000000000

X09600000

* WRITE BREF FROM ALARS WITH NO REAL SECURITY
  PCTEH LX71, PGM=(UHL0260P, UHL0400P), SERVICE=4, RMV=LI

* WRITE BREF FROM ALARS: (LAST FLAG BYTE = RLC)
  PCTEH LX72, PGM=(UHL0260P, UHL0400P), SERVICE=4, RMV=LI,

  FLAGS=00000000000000D9

X09700000

* LY HISTORY LOOK-UP BY SSN
  PCTEH LY01, PGM=(UHL0090P, UHL0060P),

  RMV=LI, DUPKEY=LD04

X09740000

* STOPPER/ENABLER HOST FUNCTION
  PCTEH L201, PGM=(UHL2999P), RMV=LI

  X09770000

* PCTEH L202, PGM=(UHL2010P, UHL2000P, UHL2020P), RMV=LI,

  SERVICE=22

X09780000

* PCTEH L203, PGM=(UHL2010P, UHL2000P, UHL2020P), RMV=UDUP,

  SERVICE=22

X09800000

* PCTEH L204, PGM=(UHL2010P, UHL2000P, UHL2030P), RMV=LI,

  SERVICE=22

X09820000

* MAB STOPPER/ENABLER HOST
  PCTEH L205, PGM=(UHL2010P, UHL2000P, UHL2040P), RMV=LI,

  SERVICE=22

X09850000

* MAB STOPPER/ENABLER HOST FOR TEST ONLY
  PCTEH L206, PGM=(UHL2010P, UHL2000S, UHL2040S), RMV=LI,

  SERVICE=22

X09880000

* MATT INQUIRE ON MATT TEST BED FILE
  PCTEH MA01, PGM=(UHT0010P), SERVICE=35, RMV=MATE

X09900000

* MATT:INCOMPLETE UPDATES TO TESTBED
  PCTEH MA02, PGM=(UHT0060P), SERVICE=35, RMV=MATE

X09910000

* MATT:UPDATE TEST BED "COMPLETE" SWITCH
  PCTEH MA03, PGM=(UHT0030P), SERVICE=35, RMV=MATE

X09920000

* MATT:UPDATE TEST BED "COMPLETE" SWITCH
Appendix C – Host Process Control Table

* MATT: RETRIEVE LIST OF TEST CASES FROM TESTBED
  PCTEH MA04, PGM=(UHT0050P), SERVICE=35, RMV=MATE
  09960000
* MATT: DELETE TESTBED RECORD
  PCTEH MA05, PGM=(UHT0040P), SERVICE=35, RMV=MATE
  09990000
* MATT: READ TEST BED THEN WRITE MATTIN & UNI TDQ RECORDS
  PCTEH MA06, PGM=(UHT0020P), SERVICE=35, RMV=MATE
  10000000
* MATT: COPY TEST BED RECORD TO CREATE NEW TEST CASE
  PCTEH MA07, PGM=(UHT0070P), SERVICE=35, RMV=MATE
  10010000
* MATT INQUIRE ON MATT OUT FILE
  PCTEH MA08, PGM=(UHT0080P), SERVICE=35, RMV=MATV
  10020000
* MATT INQUIRE ON UNI APPLICATION MESSAGE FILE
  PCTEH MA09, PGM=(UHT0090P), SERVICE=35, RMV=MATV
  10030000
* MAB ADD A COMPLAINT
  PCTEH MA20, PGM=(UHH0200P), RMV=MANC, SERVICE=6
  10040000
* MAB INFORMATION – DELETE A COMPLAINT
  PCTEH MA21, PGM=(UHH0210P), RMV=MADN
  10050000
* MAB INFORMATION – CANCEL HP/DV PLATE
  PCTEH MA22, PGM=(UHH0220P), RMV=MACP
  10060000
* MAB CANCEL PLACARD
  PCTEH MA23, PGM=(UHH0230P), RMV=MACP
  10070000
* MAB GET CUSTOMERS INFORMATION MAB DATA
  PCTEH MA24, PGM=(UHH0240P), RMV=APMN
  10080000
* MAB RETRIEVE CUSTOMER MEDICAL CONDITIONS
  PCTEH MA25, PGM=(UHH0250P), SERVICE=44, RMV=MAMI
  10090000
* MAB UPDATE CUSTOMERS MAB INFORMATION
  PCTEH MA26, PGM=(UHH0260P), RMV=MAMI
  10100000
* MAB INFORMATION – ISSUE HP/DV PLATE
  PCTEH MA27, PGM=(UHH0270P), RMV=MAMI
  10110000
* MAB INFORMATION – ISSUE HP/DV PLATE
  PCTEH MA28, PGM=(UHH0280P), RMV=MAPL
  10120000
* MAB PRINT PLACARD
  PCTEH MA29, PGM=(UHH0290P), SERVICE=28, RMV=MAPL
  10130000
* MAB NOTES INQUIRY
  PCTEH MA30, PGM=(UHH0300P), RMV=MAPL
  10140000
* MAB NOTES ADD
  PCTEH MA31, PGM=(UHH0310P), RMV=MATL
  10150000
* MAB PRINT PLACARD
  PCTEH MA32, PGM=(UHH0320P), RMV=MAOV
  10160000
* MAB RETRIEVE PLACARD TRIGGERS
  PCTEH MA33, PGM=(UHH0330P), SERVICE=44
  10170000
* MAB OVERRIDE
  PCTEH MA34, PGM=(UHH0340P), RMV=MAPL
  10180000
* MAB UPDATE PLACARD TRIGGERS
  PCTEH MA35, PGM=(UHH0350P)
  10190000
* MAB RESOLVE NOT STO COMPLAINT
  PCTEH MA36, PGM=(UHH0360P), RMV=MANC
  10200000
* MAB TRANSACTION LOG INQUIRY
  PCTEH MA37, PGM=(UHH0370P), RMV=MATL
  10210000
* MAB INFORMATION – GET HEALTHCARE PRVDR SURR BY REG#
  PCTEH MA38, PGM=(UHH0380P), RMV=MAMI
  10220000
* MAB REQUEST LETTER
  PCTEH MA39, PGM=(UHH0390P)
  10230000
* MAB IDENTIFY CUSTOMER BY PLATE TYPE AND NUMBER
  PCTEH MA40, PGM=(UHH0400P), DUPKEY=UR05, RMV=APMN
  10240000
* MAB GET ALL HEALTHCARE PROVIDERS FOR A REGISTRATION
  PCTEH MA41, PGM=(UHH0410P), RMV=MAMI
  10250000
* MAB ADD HEALTHCARE PRIVDER
  PCTEH MA42, PGM=(UHH0420P), RMV=MAMI
  10260000
* MAB INFORMATION - UNDO APPROVAL 10530000
  PCTEH MA44,PGM=(UHH0440P),RMV=MAOV
* MAB INFORMATION - MULTIPLE PLATE LIST 10550000
  PCTEH MA45,PGM=(UHH0450P),RMV=MAPL
* MAB INFORMATION - GET HEALTHCARE PRVDR DATA BY SURR 10570000
  PCTEH MA47,PGM=(UHH0470P),RMV=MAMI
* MAB RENEW, REPLACE OR EXTEND PLACARD 10590000
  PCTEH MA48,PGM=(UHH0480P),SERVICE=28,RMV=MAPL
* MAB RETRIEVE UNPRINTED PLACARD TRIGGERS 10610000
  PCTEH MA49,PGM=(UHH0490P),SERVICE=44
* MAB STOPPER LETTER CODE LOOK-UP 10630000
  PCTEH MA50,PGM=(UHH0500P)
* MAB RETRIEVE PLACARD DATA FOR PLACARD PRINTING 10650000
  PCTEH MA51,PGM=(UHH0510P)
* MAB INFORMATION - GET REFERENCE DOCUMENT DATA 10670000
  PCTEH MA52,PGM=(UHH0520P),RMV=MAMI
* MAB INFORMATION - CREATE PLACARD FOR PENDING TRIGGER RECS 10690000
  PCTEH MA53,PGM=(UHH0530P),SERVICE=44
* MAB INFORMATION - READ DISABLED PLATES 10710000
  PCTEH MA54,PGM=(UHH0540P),RMV=MAPL
* MAB - CHECK FOR EXISTING PLATE IN MAB AND ALARS 10730000
  PCTEH MA55,PGM=(UHH0550P)
* MAB - ISSUE RENEW CANCEL MEDICAL WAIVER 10750000
  PCTEH MA56,PGM=(UHH0560P)
* MAB - OBSOLETE FUNCTION 10770000
  PCTEH MA58,PGM=(UHH0580P)
* MAB - ISSUE MAB AGREEMENT 10790000
  PCTEH MA60,PGM=(UHH0600P),SERVICE=46
* MAB - UPDATE MAB AGREEMENT 10810000
  PCTEH MA61,PGM=(UHH0610P)
* MAB - CLOSE MAB AGREEMENT 10830000
  PCTEH MA62,PGM=(UHH0620P)
* MAB - DELETE MAB AGREEMENT 10850000
  PCTEH MA63,PGM=(UHH0630P)
* MAB - RETRIEVE MULTIPLE MAB AGREEMENTS 10870000
  PCTEH MA64,PGM=(UHH0640P)
* MAB - RETRIEVE A SINGLE MAB AGREEMENT 10890000
  PCTEH MA65,PGM=(UHH0650P)
* MAB - UPDATE LETTER TEXT 10910000
  PCTEH MA66,PGM=(UHH0660P)
* MRB DUPLICATE LICENSE SCROLL 10930000
  PCTEH MRBX,PGM=(UHM0002P),RMV=MO
* RETRIEVE OPERATOR LICENSE INFORMATION (MRB) 10950000
  PCTEH MRBO,PGM=(UHM0001P),DUPKEY=MRBX,RMV=MO
* MRB CUSTOMER SERVICE INQUIRY 10970000
  PCTEH MS01,PGM=(UHM0060P)
* MRB CUSTOMER SERVICE UPDATE 10990000
  PCTEH MS02,PGM=(UHM0062P)
* NMH NMTVIS MESSAGES LOG SCREEN 11010000
  PCTEH NM01,PGM=(UHR1363P),RMV=NMH
* NEW PASSWORD, PASSWORD CHANGER 11030000
  PCTEH NP20,PGM=(UHZ0018P,UHZ0105P),RMV=CMC
* NONRENEW, GENERAL ADDRESS EDIT, FULL FUNCTION 11050000
  PCTEH NRAE,PGM=UHN0160P,RMV=NRRAE
* NONRENEW, GENERAL ADDRESS EDIT, UGZ0006P ONLY 11070000
  PCTEH NRRAF,PGM=UHN0160P,RMV=NRRAE,FLAGS=0100000000000000
* NONRENEW STATUS 11090000
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCTEH NRA0, PGM=UHN0100P, RMV=NRL</td>
<td>* NONRENEW, GENERAL ADDRESS EDIT, FULL FUNCTION 40 BYTE</td>
<td>11100000</td>
</tr>
<tr>
<td>PCTEH NRBE, PGM=UHN0161P, RMV=NRAE</td>
<td></td>
<td>11120000</td>
</tr>
<tr>
<td>PCTEH NRBF, PGM=UHN0161P, RMV=NRAE, FLAGS=0100000000000000</td>
<td>* NONRENEW WARRANT INQUIRY</td>
<td>11140000</td>
</tr>
<tr>
<td>PCTEH NR18, PGM=UHN0180P</td>
<td>* NONRENEW TICKET# INQUIRY</td>
<td>11150000</td>
</tr>
<tr>
<td>PCTEH NR20, PGM=(UHN0020P, UHN0100P), RMV=NRL</td>
<td>* NONRENEW WARRANT APPLY AND CLEAR EXEMPTION</td>
<td>11190000</td>
</tr>
<tr>
<td>PCTEH NR22, PGM=(UHN0220P), RMV=LWAX</td>
<td>* NONRENEW Warrant Exemption Inquiry</td>
<td>11230000</td>
</tr>
<tr>
<td>PCTEH NR24, PGM=(UHN0240P), RMV=LWIX</td>
<td>* NONRENEW TICKET CALCULATION COST</td>
<td>11250000</td>
</tr>
<tr>
<td>PCTEH NR30, PGM=(UHN0310P)</td>
<td>* NONRENEW INDEXED TICKET INQUIRY</td>
<td>11260000</td>
</tr>
<tr>
<td>PCTEH NR40, PGM=(UHN0040P, UHN0100P), RMV=NRL</td>
<td>* NONRENEW INDEXED TICKET INQUIRY WITH TOWN/SUBSYS</td>
<td>11280000</td>
</tr>
<tr>
<td>PCTEH NR41, PGM=(UHN0040P, UHN0100P), RMV=NRL, X1</td>
<td>* NONRENEW INDEXED CLEAR TICKET INQUIRY</td>
<td>11310000</td>
</tr>
<tr>
<td>PCTEH NR42, PGM=(UHN0120P, UHN0100P), RMV=NRL</td>
<td>* NONRENEW INDEXED CLEAR TICKET INQUIRY</td>
<td>11330000</td>
</tr>
<tr>
<td>PCTEH NR50, PGM=(UHN0050P, UHN0300P), RMV=NRL</td>
<td>* NONRENEW SWAP INQUIRY</td>
<td>11350000</td>
</tr>
<tr>
<td>PCTEH NR60, PGM=(UHN0060P, UHN0300P), RMV=NRL</td>
<td>* NONRENEW UPDATE, MARK FUNCTION</td>
<td>11370000</td>
</tr>
<tr>
<td>PCTEH NR80, PGM=(UHN0080P, UHN0300P), RMV=NRM</td>
<td>* NONRENEW UPDATE, CLEAR FUNCTION</td>
<td>11390000</td>
</tr>
<tr>
<td>PCTEH NR81, PGM=(UHN0080P, UHN0300P), RMV=NRC</td>
<td>* NONRENEW UPDATE, CHANGE TICKET NUMBER FUNCTION</td>
<td>11410000</td>
</tr>
<tr>
<td>PCTEH NR82, PGM=(UHN0080P, UHN0300P), RMV=NRX</td>
<td>* NONRENEW UPDATE, MARK FUNCTION (EXTERNAL)</td>
<td>11430000</td>
</tr>
<tr>
<td>PCTEH NR83, PGM=(UHN0080P, UHN0300P), RMV=NRMX, X1</td>
<td>* NONRENEW UPDATE, CLEAR FUNCTION (EXTERNAL)</td>
<td>11450000</td>
</tr>
<tr>
<td>PCTEH NR84, PGM=(UHN0080P, UHN0300P), RMV=NRC, X1</td>
<td>* NONRENEW UPDATE, CHANGE NUMBER FUNCTION (EXTERNAL)</td>
<td>11470000</td>
</tr>
<tr>
<td>PCTEH NR85, PGM=(UHN0080P, UHN0300P), RMV=NRX, X1</td>
<td>* NONRENEW UPDATE / WARRANT EXCLUSION UPDATE</td>
<td>11490000</td>
</tr>
<tr>
<td>PCTEH NR99, PGM=(UHN0420P), RMV=NRM</td>
<td>* SOA - OOS ACCIDENTS (INQ BY MA LICENSE NUMBER)</td>
<td>11510000</td>
</tr>
<tr>
<td>PCTEH OA01, PGM=(UHS1173P), RMV=SOA, DUPKEY=LI04, SERVICE=9</td>
<td>* SOA - OOS ACCIDENTS (INQ BY OOS LICENSE NUMBER)</td>
<td>11530000</td>
</tr>
<tr>
<td>PCTEH OA02, PGM=(UHS1173P), RMV=SOA, DUPKEY=LD08, SERVICE=9</td>
<td>* SOA - OOS ACCIDENTS (UPDATE OUT OF STATE ACCIDENT)</td>
<td>11550000</td>
</tr>
<tr>
<td>PCTEH OA03, PGM=(UHS1174P), RMV=SOA</td>
<td>*</td>
<td>11570000</td>
</tr>
<tr>
<td>PCTEH OA03, PGM=(UHS1174P), RMV=SOA</td>
<td>*</td>
<td>11590000</td>
</tr>
<tr>
<td>PCTEH OA03, PGM=(UHS1174P), RMV=SOA</td>
<td>*</td>
<td>11610000</td>
</tr>
<tr>
<td>PCTEH OC01, PGM=(UHS1153P), RMV=SOA, DUPKEY=LI04, SERVICE=9</td>
<td>* SOA - OOS CONVICTIONS (INQ BY MA LICENSE NUMBER)</td>
<td>11550000</td>
</tr>
<tr>
<td>PCTEH OC02, PGM=(UHS1153P), RMV=SOA, DUPKEY=LD08, SERVICE=9</td>
<td>* SOA - OOS CONVICTIONS (INQ BY OOS LICENSE NUMBER)</td>
<td>11570000</td>
</tr>
<tr>
<td>PCTEH OC02, PGM=(UHS1153P), RMV=SOA, DUPKEY=LD08, SERVICE=9</td>
<td>* SOA - OOS CONVICTIONS (UPDATE OUT OF STATE CONVICTION)</td>
<td>11590000</td>
</tr>
<tr>
<td>PCTEH OC02, PGM=(UHS1153P), RMV=SOA, DUPKEY=LD08, SERVICE=9</td>
<td>* SOA - OOS CONVICTIONS (UPDATE OUT OF STATE CONVICTION)</td>
<td>11610000</td>
</tr>
</tbody>
</table>

**Appendix C – Host Process Control Table**
PCTEH OC03, PGM=(UHS1154P), RMV=SOC, SERVICE=9

SOC - OOS CONVICTIONS (INQ BY MA LICENSE NUMBER)

PCTEH OC04, PGM=(UHS2153P), RMV=SOC, DUPKEY=LI04, SERVICE=9

PCTEH OC05, PGM=(UHS2153P), RMV=SOC, DUPKEY=LD08, SERVICE=9

SOC - OOS CONVICTIONS (UPDATE OUT OF STATE CONVICTION)

PCTEH OC06, PGM=(UHS2154P), RMV=SOC, SERVICE=9

* SOC - OOS CONVICTIONS (INQ BY MA LICENSE NUMBER)

PCTEH OC07, PGM=(UHS3153P), RMV=SOC, DUPKEY=LI04, SERVICE=9

PCTEH OC08, PGM=(UHS3153P), RMV=SOC, DUPKEY=LD08, SERVICE=9

SOC - OOS CONVICTIONS (UPDATE OUT OF STATE CONVICTION)

PCTEH OC09, PGM=(UHS3154P), RMV=SOC, SERVICE=9

SOW - OOS WITHDRAWALS (INQ BY MA LICENSE NUMBER)

PCTEH OW01, PGM=(UHS1163P), RMV=SOW, DUPKEY=LI04, SERVICE=9

PCTEH OW02, PGM=(UHS1163P), RMV=SOW, DUPKEY=LD08, SERVICE=9

* SOW - OOS WITHDRAWALS (INQ BY OOS LICENSE NUMBER)

PCTEH OW03, PGM=(UHS1164P), RMV=SOW

OV - COMMERCIAL VEHICLE OVERWEIGHT INQUIRY

PCTEH OW04, PGM=(UHW1143Q), RMV=OVW

OWH - COMMERCIAL VEHICLE OVERWEIGHT HISTORY

PCTEH OW05, PGM=(UHW1133Q), RMV=OWH

UPA - POLICY AMEND (INQ BY POLICY ID) (NEW)

PCTEH PA05, PGM=(UHU2123P), RMV=UPA, SERVICE=1

PCTEH PA06, PGM=(UHU2123P), RMV=UPA, DUPKEY=LI04, SERVICE=1

* UPA - POLICY AMEND (INQ BY POLICY ID) (NEW)

PCTEH PA07, PGM=(UHU2123P), RMV=UPA, DUPKEY=LD08, SERVICE=1

* UPA - POLICY AMEND (INQ BY MA LICENSE NUMBER) (NEW)

PCTEH PA08, PGM=(UHU2124P), RMV=UPA, SERVICE=1

* UPA - POLICY AMEND (INQ BY FID)

PCTEH PA10, PGM=(UHU3123P), RMV=UPA, SERVICE=1

PCTEH PA11, PGM=(UHU3123P), RMV=UPA, DUPKEY=LI04, SERVICE=1

* UPA - POLICY AMEND (INQ BY MA LICENSE NUMBER)

PCTEH PA12, PGM=(UHU3123P), RMV=UPA, DUPKEY=LD08, SERVICE=1

* UPA - POLICY AMEND (UPDATE POLICY INFO)

PCTEH PA13, PGM=(UHU3124P), RMV=UPA, SERVICE=1

* UPA - POLICY AMEND (UPDATE POLICY INFO)

PCTEH PA14, PGM=(UHU3123P), RMV=UPA, DUPKEY=UR09, SERVICE=1

CPB - BACKOUT TRANSACTIONS

PCTEH PB01, PGM=(UHC1063P), RMV=CPB

* CPG - CASH PROCESS - GENERAL COLLECTION (INQ BY MA LICENSE NUMBER)

PCTEH PG01, PGM=(UHC1053P), RMV=CPG, DUPKEY=LI04, SERVICE=52

* CPG - CASH PROCESS - GENERAL COLLECTION (INQ BY OOS LICENSE NUMBER)

PCTEH PG02, PGM=(UHC1053P), RMV=CPG, DUPKEY=LD08, SERVICE=52

* CPG - CASH PROCESS - GENERAL COLLECTION (INQ BY REG)

PCTEH PG03, PGM=(UHC1053P), RMV=CPG, DUPKEY=UR05, SERVICE=52

* CPG - CASH PROCESS - GENERAL COLLECTION (UPDATE PROCESS)

PCTEH PG04, PGM=(UHC1054P), RMV=CPG, SERVICE=52

OVERWEIGHT REDUCIBLE PERMIT ISSUANCE SCREEN

PCTEH PG05, PGM=(UHC1053P), RMV=CPG, SERVICE=52

Appendix C – Host Process Control Table
<table>
<thead>
<tr>
<th>Action Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>* OW - PERMIT ISSUE (INQ BY REG)</td>
<td></td>
</tr>
<tr>
<td>PCTEH P101, PGM=(UHW1113P), RMV=OW, DUPKEY=UR05</td>
<td>12270000</td>
</tr>
<tr>
<td>* OW - PERMIT ISSUE (INQ BY VIN)</td>
<td></td>
</tr>
<tr>
<td>PCTEH P102, PGM=(UHW1113P), RMV=OW, DUPKEY=UR03</td>
<td>12280000</td>
</tr>
<tr>
<td>* OW - PERMIT ISSUE (LIC INQUIRY)</td>
<td></td>
</tr>
<tr>
<td>PCTEH P103, PGM=(UHW1113P), RMV=OW, DUPKEY=LI04</td>
<td>12300000</td>
</tr>
<tr>
<td>* UPDATE PASSENGER RESTRICTION DATA</td>
<td></td>
</tr>
<tr>
<td>PCTEH P02, PGM=(UHL3202P), RMV=LPRM, SERVICE=18</td>
<td>12310000</td>
</tr>
<tr>
<td>* RETRIEVE PASSENGER RESTRICTION DATA</td>
<td></td>
</tr>
<tr>
<td>PCTEH P04, PGM=(UHL3204P)</td>
<td>12320000</td>
</tr>
<tr>
<td>* PASSENGER RESTRICTION INQUIRY</td>
<td></td>
</tr>
<tr>
<td>PCTEH P05, PGM=(UHL3205P)</td>
<td>12330000</td>
</tr>
<tr>
<td>* JOL STATUS INQUIRY</td>
<td></td>
</tr>
<tr>
<td>PCTEH P06, PGM=(UHL3206P)</td>
<td>12340000</td>
</tr>
<tr>
<td>* RA - REG AMEND (INQ BY REG)</td>
<td></td>
</tr>
<tr>
<td>PCTEH R01, PGM=(UHR1273P), RMV=RA, DUPKEY=UR05, SERVICE=1</td>
<td>X12350000</td>
</tr>
<tr>
<td>* RA - REG AMEND (INQ BY VIN)</td>
<td></td>
</tr>
<tr>
<td>PCTEH R02, PGM=(UHR1273P), RMV=RA, DUPKEY=UR03, SERVICE=1</td>
<td>X12360000</td>
</tr>
<tr>
<td>* RA - REG AMEND (LIC INQUIRY)</td>
<td></td>
</tr>
<tr>
<td>PCTEH R03, PGM=(UHR1273P), RMV=RA, DUPKEY=LI04, SERVICE=1</td>
<td>X12370000</td>
</tr>
<tr>
<td>* STOLEN VEHICLE UPDATES (STOLEN/RECOVERED VEHICLE/PLATES)</td>
<td>12380000</td>
</tr>
<tr>
<td>PCTEH R02, PGM=(UHR1293P, UHR1294P), RMV=RC</td>
<td>12390000</td>
</tr>
<tr>
<td>* REGISTRATION HISTORY INQUIRY</td>
<td></td>
</tr>
<tr>
<td>PCTEH R01, PGM=(UHR1253P), RMV=RH, SERVICE=9</td>
<td>12400000</td>
</tr>
<tr>
<td>* REGISTRATION STORE/MODIFY/ERASE MMVR-VEHR-ADDR</td>
<td></td>
</tr>
<tr>
<td>PCTEH R120, PGM=(UHR0020P)</td>
<td>12410000</td>
</tr>
<tr>
<td>* REGISTRATION SCROLL BY NUMBER</td>
<td></td>
</tr>
<tr>
<td>PCTEH R01, PGM=(UHR0010P), RMV=RN, SERVICE=1</td>
<td>12420000</td>
</tr>
<tr>
<td>* RP - VANITY/SPECIAL PLATE ORDER</td>
<td></td>
</tr>
<tr>
<td>PCTEH R00, PGM=(UHR0313P), RMV=RP</td>
<td>12430000</td>
</tr>
<tr>
<td>* REGISTRATION SUPERQUERY. SC0197.</td>
<td></td>
</tr>
<tr>
<td>PCTEH R01, PGM=(UHR0040P, UHR0060P), RMV=RS</td>
<td>12440000</td>
</tr>
<tr>
<td>* R1A - RMV-1 RX/SW ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R01, PGM=(UHR1183P), RMV=R1A, DUPKEY=LI03</td>
<td>12450000</td>
</tr>
<tr>
<td>* R1A - RMV-1 RO/RT ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R02, PGM=(UHR1184P), RMV=R1A, DUPKEY=LI03</td>
<td>12460000</td>
</tr>
<tr>
<td>* R1A - RMV-1 TO/ST ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R03, PGM=(UHR1185P), RMV=R1A, DUPKEY=LI03</td>
<td>12470000</td>
</tr>
<tr>
<td>* R1A - RMV-1 RX/SW ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R04, PGM=(UHR1283P), RMV=R1A, SERVICE=1 EDIT=FORCE</td>
<td>12480000</td>
</tr>
<tr>
<td>* R1A - RMV-1 RO/RT ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R05, PGM=(UHR1284P), RMV=R1A, SERVICE=1 EDIT=FORCE</td>
<td>12490000</td>
</tr>
<tr>
<td>* R1A - RMV-1 TO/ST ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R06, PGM=(UHR1285P), RMV=R1A, SERVICE=1 EDIT=FORCE</td>
<td>12500000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW/RO/RT/TDO/ST ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R07, PGM=(UHR1295P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12510000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW RO/RT/TDO/SS ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R08, PGM=(UHR1305P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12520000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW/RO/RT/TO/ST/SS ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R09, PGM=(UHR1315P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12530000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW RO/RT/TO/ST/SS/SS ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R10, PGM=(UHR1325P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12540000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW RO/RT/TO/ST/SS/SS/SS ACTIONS</td>
<td></td>
</tr>
<tr>
<td>PCTEH R11, PGM=(UHR1335P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12550000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW RO/RT/TO/ST/SS/SS/SS Actions</td>
<td></td>
</tr>
<tr>
<td>PCTEH R12, PGM=(UHR1345P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12560000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW RO/RT/TO/ST/SS/SS/SS Actions</td>
<td></td>
</tr>
<tr>
<td>PCTEH R13, PGM=(UHR1355P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12570000</td>
</tr>
<tr>
<td>* R1B - RMV-1 RX/SW RO/RT/TO/ST/SS/SS/SS Actions</td>
<td></td>
</tr>
<tr>
<td>PCTEH R14, PGM=(UHR1365P), RMV=R1B, SERVICE=1 EDIT=FORCE</td>
<td>12580000</td>
</tr>
</tbody>
</table>
* R1A - RMV-1 SS ACTION
  * PCTEH R115, PGM=(UHR2187P), RMV=R1A, EDIT=FORCE
  * PCTEH R116, PGM=(UHR2303P), RMV=R1C
* EVRR - EVR RESEND RESPONSE SCREEN
  * PCTEH R117, PGM=(UHR1173P), RMV=EVRR
* SUSPENSIONS DRIVER HISTORY INQUIRY (BY MA LICENSE NUMBER)
  * PCTEH SDH00, PGM=(UHL0070P, UHL0100P), SERVICE=6,
    * RMV=SDH, DUPKEY=LI04
    * X12960000
* SUSPENSIONS DRIVER HISTORY INQUIRY (BY OOS LICENSE NUMBER)
  * PCTEH SDH11, PGM=(UHL0080P, UHL0100P), SERVICE=6,
    * RMV=SDH, DUPKEY=LD08
    * X12990000
* SUSPENSIONS DRIVER HISTORY INQUIRY (BY SOCIAL SECURITY NO)
  * PCTEH SDH2, PGM=(UHL0090P, UHL0100P), SERVICE=6,
    * RMV=SDH, DUPKEY=LD04
    * X13020000
* SUSPENSIONS DRIVER HISTORY INQUIRY (PERSON SURROGATE)
  * PCTEH SDH3, PGM=(UHL0100P), SERVICE=6,
    * RMV=SDH
    * X13050000
* SUSPENSIONS DRIVER HISTORY INQUIRY (OBTAINING UNOP DATA)
  * PCTEH SDH4, PGM=(UHS1103P), SERVICE=6,
    * RMV=SDH
    * X13080000
* SUSPENSIONS DRIVER HISTORY INQUIRY (READ OUTBOUND TRANSFER FILE)
  * PCTEH SDH5, PGM=(UHS1104P), SERVICE=6,
    * RMV=SDH
    * X13110000
* PRINT DEMOGRAPHICS SUB-FUNCTION SECURITY
  * PCTEH SEC0, PGM=(UHL1120P)
    * X13140000
* AFW MAIN MENU SECURITY
  * PCTEH SEC1, PGM=(UHR1993P), SERVICE=1, RMV=RI
    * X13170000
* SUSPENSION HISTORY BY LIC NUMBER
  * PCTEH SH01, PGM=(UHS1100P, UHS0100P), RMV=SH, DUPKEY=LI04,
    * SERVICE=2
    * X13230000
* SUSPENSION HISTORY BY SURROGATE
  * PCTEH SH02, PGM=(UHS0100P), RMV=SH
    * SERVICE=2
    * X13260000
* SUSPENSION HISTORY BY LIC NUMBER (NEW)
  * PCTEH SH03, PGM=(UHL0070P, UHS1054P), RMV=SH, DUPKEY=LI04,
    * SERVICE=25
    * X13320000
* SUSPENSION HISTORY BY SURROGATE (NEW)
  * PCTEH SH04, PGM=(UHS1054P), RMV=SH
    * SERVICE=25
    * X13350000
* SUSPENSION HISTORY BY OOS NUMBER (NEW)
  * PCTEH SH05, PGM=(UHL0080P, UHS1054P), RMV=SH, DUPKEY=LD08,
    * SERVICE=25
    * X13370000
* END OF REMOVAL OF NO LONGER USED SH FUNCTIONS
  * X13390000
* SUSPENSION HISTORY BY LIC NUMBER (NEW)
  * PCTEH SH06, PGM=(UHL0070P, UHS2054P), RMV=SH, DUPKEY=LI04,
    * SERVICE=9
    * X13430000
* SUSPENSION HISTORY BY SURROGATE (NEW)
  * PCTEH SH07, PGM=(UHS2054P), RMV=SH
    * SERVICE=9
    * X13460000
* SUSPENSION HISTORY BY OOS NUMBER (NEW)
  * PCTEH SH08, PGM=(UHL0080P, UHS2054P), RMV=SH
    * DUPKEY=LD08
    * SERVICE=9
    * X13480000
* SUSPENSION HISTORY BY REG (NEW)                                        13500000
  PCTEH SH09, PGM=(UHS2053P, UHS2054P), RMV=SH, DUPKEY=UR05,          X13510000
  SERVICE=9                                                      13520000
* SUSPENSION HISTORY BY LIC NUMBER (NEW)                                13530000
  PCTEH SH10, PGM=(UHL0070P, UHS3054P), RMV=SH, DUPKEY=LI04,          X13540000
  SERVICE=9                                                      13550000
* SUSPENSION HISTORY BY SURROGATE (NEW)                                 13560000
  PCTEH SH11, PGM=(UHS3054P), RMV=SH, SERVICE=9                        13570000
* SUSPENSION HISTORY BY OOS NUMBER (NEW)                                 13580000
  PCTEH SH12, PGM=(UHL0080P, UHS3054P), RMV=SH, DUPKEY=LD08,          X13590000
  SERVICE=9                                                      13600000
* SUSPENSION HISTORY BY REG (NEW)                                        13610000
  PCTEH SH13, PGM=(UHS3053P, UHS3054P), RMV=SH, DUPKEY=UR05,          X13620000
  SERVICE=9                                                      13630000
* SCHEDULE MAINTENANCE CALENDAR INQUIRY                                 13640000
  PCTEH SMC1, PGM=(UHB2200P), FLAGS=C9000000000000000 , RMV=          13650000
* SCHEDULE MAINTENANCE CALENDAR UPDATE                                   13660000
  PCTEH SMCU, PGM=(UHB2200P), FLAGS=E4000000000000000 , RMV=          13670000
* SSOLV OUTBOUND INQUIRY HOST                                           13680000
  PCTEH SS01, PGM=(UHL3400P)                                          13690000
* SSOLV - RETRIEVE SSA RESPONSE FROM RAD FILE                           13700000
  PCTEH SS02, PGM=(UHL0870P)                                          13710000
* SSOLV - RESEND PROCESSING                                             13720000
  PCTEH SS03, PGM=(UHL3430P)                                          13730000
* SSOLV - SSN STATUS HOST                                               13740000
  PCTEH SS04, PGM=(UHL3440P)                                          13750000
* ADDED 05/02/96                                                         13760000
* SUSPENSIONS OUI UPDATES (INSTATE ADM AND CTR)                         13770000
  PCTEH SU01, PGM=(UHS1183P, UHS1184P), RMV=SO                      13780000
* UTAS TITLE AUDIT SCREEN                                               13790000
  PCTEH TAS1, PGM=UHR1083P, RMV=TAS                                    13800000
* ALOG HISTORY INQUIRY BY REGISTRATION                                  13810000
  PCTEH TX01, PGM=(UHR1283P), RMV=TXH, DUPKEY=UR05                    13820000
* ALOG HISTORY INQUIRY BY TITLE                                         13830000
  PCTEH TX02, PGM=(UHR1283P), RMV=TXH                                  13840000
  13850000
* TIA - TITLE AMEND (INQ BY TITLE/BATCH NUMBER - UPDATE TITLE)           13860000
  PCTEH TIA1, PGM=(UHR1213P), RMV=TIA, SERVICE=1                     13870000
* TIA - TITLE AMEND (INQ BY REG)                                         13880000
  PCTEH TIA2, PGM=(UHR1213P), RMV=TIA, DUPKEY=UR05, SERVICE=1         13890000
* TIA - TITLE AMEND (INQ BY VIN)                                        13900000
  PCTEH TIA3, PGM=(UHR1213P), RMV=TIA, DUPKEY=UR03, SERVICE=1          13910000
* TIA - TITLE AMEND (LIEN INQUIRY)                                      13920000
  PCTEH TIA4, PGM=(UHR1213P), RMV=TIA, DUPKEY=BS02, SERVICE=1         13930000
* TIB - TITLE AMEND CASH (FTAB INQUIRY - UPDATE TITLE)                   13940000
  PCTEH TIB1, PGM=(UHR1223P), RMV=TIB                                13950000
  PCTEH TIB1, PGM=(UHR1223P), RMV=TIB                                  13960000
* TIC - ALTERNATE ADDRESS INFORMATION                                    13970000
  PCTEH TIC1, PGM=(UHR1313P), RMV=TIC                                 13980000
* MRB AT-FAULT CLAIM INQUIRY                                            13990000
  PCTEH UMA1, PGM=(UHM0030P), RMV=MA                                  14000000
* MRB ADMINISTRATIVE DETAIL                                              14010000
  PCTEH UMB1, PGM=(UHM0150P), RMV=MB                                   14020000
* MRB CITATION BATCH CONTROL                                             14030000
  PCTEH UMB2, PGM=(UHM0151P), RMV=MV                                  14040000
* MRB COMPREHENSIVE CLAIM INQUIRY                                       14050000
  PCTEH UMC1, PGM=(UHM0040P), RMV=MC                                  14060000
<table>
<thead>
<tr>
<th>Procedure Description</th>
<th>Service Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>* MRB DRIVING HISTORY</td>
<td>14070000</td>
</tr>
<tr>
<td>PCTEH UMDX, PGM=(UHM0090P), RMV=MDH</td>
<td>14080000</td>
</tr>
<tr>
<td>* MRB ADMINISTRATIVE DETAIL</td>
<td>14090000</td>
</tr>
<tr>
<td>PCTEH UMDY, PGM=(UHM0094P), RMV=MDD</td>
<td>14100000</td>
</tr>
<tr>
<td>* MRB RETRIEVE CURRENT SDIP INFORMATION</td>
<td>14110000</td>
</tr>
<tr>
<td>PCTEH UMI1, PGM=(UHM0020P), RMV=M19</td>
<td>14120000</td>
</tr>
<tr>
<td>* MRB RETRIEVE SDIP INQUIRY HISTORY</td>
<td>14130000</td>
</tr>
<tr>
<td>PCTEH UMI2, PGM=(UHM0021P), RMV=M19</td>
<td>14140000</td>
</tr>
<tr>
<td>* MRB SDIP INQUIRY UPDATE</td>
<td>14150000</td>
</tr>
<tr>
<td>PCTEH UMI3, PGM=(UHM0022P), RMV=M19</td>
<td>14160000</td>
</tr>
<tr>
<td>* MRB NO CREDIT PERIOD MAINTENANCE</td>
<td>14170000</td>
</tr>
<tr>
<td>PCTEH UMNC, PGM=(UHM0012P)</td>
<td>14180000</td>
</tr>
<tr>
<td>* RETRIEVE OPERATOR SUMMARY INFORMATION (MRB)</td>
<td>14190000</td>
</tr>
<tr>
<td>PCTEH UMO1, PGM=(UHM0010P), RMV=MO</td>
<td>14200000</td>
</tr>
<tr>
<td>* MRB DUPLICATE LICENSE SCROLL</td>
<td>14210000</td>
</tr>
<tr>
<td>PCTEH UMVX, PGM=(UHM0055P), RMV=MVI</td>
<td>14220000</td>
</tr>
<tr>
<td>* MRB CITATION INQUIRY</td>
<td>14230000</td>
</tr>
<tr>
<td>PCTEH UMI1, PGM=(UHM0050P), DUPKEY=UMVX, RMV=MVI</td>
<td>14240000</td>
</tr>
<tr>
<td>* MRB CITATION UPDATE</td>
<td>14250000</td>
</tr>
<tr>
<td>PCTEH UMI2, PGM=(UHM0250P), DUPKEY=UMVX, RMV=MV</td>
<td>14260000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY REG NUMBER (NEW)</td>
<td>14270000</td>
</tr>
<tr>
<td>PCTEH URIA, PGM=(UHR3063P, UHR3067P), RMV=RI, DUPKEY=UR05, SERVICE=1</td>
<td>14280000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY VIN NUMBER (NEW)</td>
<td>14300000</td>
</tr>
<tr>
<td>PCTEH URIB, PGM=(UHR3064P, UHR3067P), RMV=RI, DUPKEY=UR03, SERVICE=1</td>
<td>14310000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY LIC NUMBER (NEW)</td>
<td>14320000</td>
</tr>
<tr>
<td>PCTEH URIC, PGM=(UHR3065P, UHR3067P), RMV=RI, DUPKEY=LI04, SERVICE=1</td>
<td>14340000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY TITLE NUMBER (NEW)</td>
<td>14350000</td>
</tr>
<tr>
<td>PCTEH URIID, PGM=(UHR3066P, UHR3067P), RMV=RI, SERVICE=1</td>
<td>14370000</td>
</tr>
<tr>
<td>* ADD VER 3 HOST FUNCTION FOR RI INQUIRY BY SURR</td>
<td>14380000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY SURROGATE NUMBER</td>
<td>14400000</td>
</tr>
<tr>
<td>PCTEH URIE, PGM=(UHR3067P), RMV=RI, SERVICE=1</td>
<td>14410000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY REG NUMBER (VERSION 4)</td>
<td>14440000</td>
</tr>
<tr>
<td>PCTEH URIF, PGM=(UHR4063P, UHR4067P), RMV=RI, DUPKEY=UR05, SERVICE=1</td>
<td>14450000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY VIN NUMBER (VERSION 4)</td>
<td>14470000</td>
</tr>
<tr>
<td>PCTEH URIG, PGM=(UHR4064P, UHR4067P), RMV=RI, DUPKEY=UR03, SERVICE=1</td>
<td>14490000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY LIC NUMBER (VERSION 4)</td>
<td>14500000</td>
</tr>
<tr>
<td>PCTEH URIH, PGM=(UHR4065P, UHR4067P), RMV=RI, DUPKEY=LI04, SERVICE=1</td>
<td>14520000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY TITLE NUMBER (VERSION 4)</td>
<td>14540000</td>
</tr>
<tr>
<td>PCTEH URII, PGM=(UHR4066P, UHR4067P), RMV=RI, SERVICE=1</td>
<td>14550000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY SURROGATE NUMBER (VERSION 4)</td>
<td>14570000</td>
</tr>
<tr>
<td>PCTEH URIJ, PGM=(UHR4067P), RMV=RI, SERVICE=1</td>
<td>14580000</td>
</tr>
<tr>
<td>* REGISTRATION INQUIRY BY REG NUMBER (VERSION 5)</td>
<td>14590000</td>
</tr>
<tr>
<td>PCTEH URIK, PGM=(UHR5063P, UHR5067P), RMV=RI, DUPKEY=UR05, X14630000</td>
<td></td>
</tr>
</tbody>
</table>

Appendix C – Host Process Control Table

257
SERVICE=1

* REGISTRATION INQUIRY BY VIN NUMBER (VERSION 5)
  PCTEH URI1, PGM=(UHR5064P, UHR5067P), RMV=RI, DUPKEY=UR03,
  SERVICE=1

* REGISTRATION INQUIRY BY LIC NUMBER (VERSION 5)
  PCTEH URI1, PGM=(UHR5065P, UHR5067P), RMV=RI, DUPKEY=L104,
  SERVICE=1

* REGISTRATION INQUIRY BY TITLE NUMBER (VERSION 5)
  PCTEH URI1, PGM=(UHR5066P, UHR5067P), RMV=RI, SERVICE=1

* REGISTRATION INQUIRY BY SURROGATE NUMBER (VERSION 5)
  PCTEH URI1, PGM=(UHR5067P), RMV=RI, SERVICE=1

* REGISTRATION INQUIRY BY REG NUMBER (VERSION 6)
  PCTEH URI1, PGM=(UHR6063P, UHR6067P), RMV=RI, DUPKEY=UR05,
  SERVICE=1

* REGISTRATION INQUIRY BY VIN NUMBER (VERSION 6)
  PCTEH URI1, PGM=(UHR6064P, UHR6067P), RMV=RI, DUPKEY=UR03,
  SERVICE=1

* REGISTRATION INQUIRY BY LIC NUMBER (VERSION 6)
  PCTEH URI1, PGM=(UHR6065P, UHR6067P), RMV=RI, DUPKEY=LI04,
  SERVICE=1

* REGISTRATION INQUIRY BY TITLE NUMBER (VERSION 6)
  PCTEH URI1, PGM=(UHR6066P, UHR6067P), RMV=RI, SERVICE=1

* REGISTRATION INQUIRY BY SURROGATE NUMBER (VERSION 6)
  PCTEH URI1, PGM=(UHR6067P), RMV=RI, SERVICE=1

* REGISTRATION INQUIRY BY REG NUMBER (NEW)
  PCTEH URI1, PGM=(UHR1063P, UHR1067P), RMV=RI, DUPKEY=UR05,
  SERVICE=1

* REGISTRATION INQUIRY BY VIN NUMBER (NEW)
  PCTEH URI1, PGM=(UHR1064P, UHR1067P), RMV=RI, DUPKEY=UR03,
  SERVICE=1

* REGISTRATION INQUIRY BY LIC NUMBER (NEW)
  PCTEH URI1, PGM=(UHR1065P, UHR1067P), RMV=RI, DUPKEY=L104,
  SERVICE=1

* REGISTRATION INQUIRY BY TITLE NUMBER (NEW)
  PCTEH URI1, PGM=(UHR1066P, UHR1067P), RMV=RI, SERVICE=1

* REGISTRATION INQUIRY BY REG NUMBER
  PCTEH URI1, PGM=(UHR1063P, UHR1067P), RMV=RI, DUPKEY=UR05,
  SERVICE=1
* REGISTRATION INQUIRY BY VIN NUMBER
  PCTEH URI6,PGM=(UHR1064P,UHR1067P),RMV=RI,DUPKEY=UR03, SERVICE=1
  X15220000 15230000
* REGISTRATION INQUIRY BY LIC NUMBER
  PCTEH URI7,PGM=(UHR1065P,UHR1067P),RMV=RI,DUPKEY=LI04, SERVICE=1
  X15250000 15260000
* REGISTRATION INQUIRY BY TITLE NUMBER
  PCTEH URI8,PGM=(UHR1066P,UHR1067P),RMV=RI,SERVICE=1
  X15280000
* REGISTRATION INQUIRY BY SURROGATE NUMBER (NEW)
  PCTEH URI9,PGM=(UHR1067P),RMV=RI,SERVICE=1
  X15300000
* REGISTRATION SCROLL
  PCTEH UR02,PGM=UHR100P,RMV=RS,SERVICE=1
  X15320000
* REGISTRATION SCROLL
  PCTEH UR03,PGM=UHR101P,RMV=RS,SERVICE=1
  X15340000
* REGISTRATION SCROLL
  PCTEH UR04,PGM=UHR102P,RMV=RS,SERVICE=1
  X15360000
* REGISTRATION SCROLL (NEW)
  PCTEH UR05,PGM=UHR1243P,RMV=RS,SERVICE=1
  X15380000
* REGISTRATION/VIN OWNER NAME SCROLL
  PCTEH UR07,PGM=UHR1093P,RMV=RS,SERVICE=1
  X15400000
**** CHANGE FOR EDS 11/17/92
* URVN - NADA INQUIRY (INQ BY VIN)
  PCTEH UR08,PGM=(UHR1143P),RMV=URVN
  X15420000
* URVN - NADA INQUIRY (INQ BY VIN)
  PCTEH UR08,PGM=(UHR1143P),RMV=RVN
  X15440000
**** END CHANGE FOR EDS 11/17/92
* RNF - CORPORATION SCROLL BY FID
  PCTEH UR09,PGM=(UHR1233P),RMV=RNF
  X15460000
* RVN
  PCTEH UR10,PGM=(UHR1243P),RMV=RVN
  X15480000
* NMP - NMVTIS PROBLEM RESOLUTION
  PCTEH UR11,PGM=(UHR1343P),RMV=NMP
  X15500000
* NMD - NMVTIS DUPLICATE VIN RESOLUTION
  PCTEH UR12,PGM=(UHR1353P),RMV=NMD
  X15520000
* POLICY HISTORY INQUIRY BY LIC NUMBER
  PCTEH UU02,PGM=(UHU110P),RMV=PH,DUPKEY=LI04,SERVICE=2
  X15540000
* TPIC
  PCTEH UU14,PGM=UHU3023P,RMV=PIC,SERVICE=3
  X15560000
* TPIC BY SURROGATE
  PCTEH UU15,PGM=UHU3024P,RMV=PIC,SERVICE=3
  X15580000
* POLICY INQUIRY BY LIC NUMBER
  PCTEH UU16,PGM=(UHU2043P),RMV=PH,DUPKEY=LI04,SERVICE=7
  X15600000
* POLICY INQUIRY BY OOS NUMBER
  PCTEH UU17,PGM=(UHU2043P),RMV=PH,DUPKEY=LD08,SERVICE=7
  X15620000
* UPOI POLICY OPERATOR INQUIRY
  PCTEH UU18,PGM=(UHU2013P,UHU2014P),RMV=POI,SERVICE=7
  X15640000
* POLICY HISTORY INQUIRY BY POLICY NUMBER
  PCTEH UU19,PGM=(UHU1133P),RMV=PTH
  X15660000
* UPIC
  PCTEH UU20,PGM=UHU4023P,RMV=PIC,SERVICE=3
  X15680000
* UPIC BY SURROGATE
  PCTEH UU21,PGM=UHU4024P,RMV=PIC,SERVICE=3
  X15700000
* POLICY INQUIRY BY LIC NUMBER
  PCTEH UU22, PGM=(UHU3043P), RMV=PH, DUPKEY=LI04, SERVICE=29
* POLICY INQUIRY BY OOS NUMBER
  PCTEH UU23, PGM=(UHU3043P), RMV=PH, DUPKEY=LD08, SERVICE=29
* POLICY INQUIRY BY FID
  PCTEH UU24, PGM=(UHU3043P), RMV=PH, DUPKEY=UR09, SERVICE=29
* UPOI POLICY OPERATOR INQUIRY
  PCTEH UU25, PGM=(UHU3013P), RMV=PH, DUPKEY=POI, SERVICE=29
* POLICY HISTORY INQUIRY BY POLICY NUMBER
  PCTEH UU26, PGM=(UHU2133P), RMV=PH
* VEHICLE POLICY HISTORY INQUIRY (BY REG NUMBER)
  PCTEH VH04, PGM=(UHU2113P), RMV=UVH, DUPKEY=UR02
  PCTEH VH04, PGM=(UHU2113P), RMV=UVH, DUPKEY=UR05
* VEHICLE POLICY HISTORY INQUIRY (BY VIN)
  PCTEH VH05, PGM=(UHU2113P), RMV=UVH, DUPKEY=UR03
* VEHICLE POLICY HISTORY INQUIRY (BY REF)
  PCTEH VH06, PGM=(UHU2113P), RMV=UVH
* RETRIEVE RAD VSAM DATA
  PCTEH VR56, PGM=(UHR0560P), RMV=LI, SERVICE=32
* UPDATE RAD VSAM FILE
  PCTEH VR57, PGM=(UHR0570P), RMV=LI, SERVICE=32
* WRITE TDQ TO START THE GENERIC OUTBOUND MESSAGE PROCESSOR
  PCTEH VR58, PGM=(UHR0580P), RMV=LI
* VEHICLE TITLE INQUIRY - OLD
  PCTEH VT01, PGM=(UHR1263P), RMV=VT, DUPKEY=UR03
* VEHICLE TITLE INQUIRY - NEW
  PCTEH VT02, PGM=(UHR2263P), RMV=VT, DUPKEY=UR03
* VEHICLE TITLE INQUIRY - NEW + 1
  PCTEH VT03, PGM=(UHR3263P), RMV=VT, DUPKEY=UR03
* TEST HOST BASED IMAGE RETREIVAL........
  PCTEH XXXX, PGM=(UHL1010P), RMV=LZI1, SERVICE=13, flags=C3000000000000000

LAST
DC XL4'FFFFFFFF'
ORG NUMENT
DC A((LAST-FIRST)/(SYM2-SYM1))
ORG
DC CL8'&SYSDATE', CL1 ' ', CL5'&SYSTIME'
END

Appendix C – Host Process Control Table
Appendix D:

CICS Miscellaneous

CICS Abend Codes List

The following is a list of the CICS abends issued by the UMS Guest programs and a brief statement of the reason why each was issued.

<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UG24</td>
<td>Invalid Function Code given</td>
</tr>
<tr>
<td>UGCA</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0005P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0006P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0007P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0008P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0009P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0011P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0013P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0022P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0023P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0024P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0025P</td>
<td>Guest common area length error</td>
</tr>
<tr>
<td>UGZ0026P</td>
<td>Guest common area length error</td>
</tr>
</tbody>
</table>
UGTH  UGZ0006P  Bad transfer of Table from Host
Transaction UG04 has failed with abend UGTH
*See note.

UGZ0024P  Read Queue Error or Load Failed for
UGZ0017p or UGZ0018p

UGTL  UGZ0001P  UGTL Txn. not run (to completion)

UMFE  UGZ0001P  Program-ID error

UGZ0005P  Error in High Core Table search

UGZ0013P  Program-ID error

UMFL  UGZ0013P

XFER  UGZ0001P  Start issued from external application
with Data, data unrecoverable.

CICS Queue Names

Queue names are usually of the form <qpx><EIBTRMID> to show the queue function
and provide uniqueness.

<qpx> Program

LIC$ UGZ0005P

UMSQ UGZ0008P

UMSR UGZ0001P  Reference List data

UGZ0007P

UGZ0022P

UMSQ UGZ0008P
A few queues have ‘complete’ names as follows:

<table>
<thead>
<tr>
<th>Queue Name</th>
<th>Complete Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSL</td>
<td>UGZ0011P</td>
<td>Log for LU6.2 messages</td>
</tr>
<tr>
<td></td>
<td>(default name)</td>
<td></td>
</tr>
<tr>
<td>MRMXDTBL</td>
<td>UGZ0026P</td>
<td></td>
</tr>
<tr>
<td>UMSGCORT</td>
<td>UGZ0024P</td>
<td>Court Table data</td>
</tr>
<tr>
<td></td>
<td>UGZ0026P</td>
<td></td>
</tr>
<tr>
<td>UMSGVTAB</td>
<td>UGZ0024P</td>
<td>Violation Table data</td>
</tr>
<tr>
<td></td>
<td>UGZ0026P</td>
<td></td>
</tr>
</tbody>
</table>

*Note: Abend UGTH with transaction UG04 is corrected by running the following utility transactions in succession:

**UGCL** *(message back to the screen will read)*
SYSTEM HAS BEEN RESET AND UGTL CAN BE RUN

**UGTL** *(message back to the screen will read)*
UGZ0020P, TABLE LOAD INITIATED
UGZ0020P, MAX=######### REMAINING=######### (where # = varying numbers)
UGZ0020P, LOAD COMPLETE
Appendix E: UMS System Programs

Guest-Side Programs

UGZ0001P GUEST CONTROL/DISPATCH PROGRAM
UGZ0002P GUEST PROCESS CONTROL TABLE
UGZ0003P GUEST UTILITY ROUTINES, RESIDENT
UGZ0004P ERROR MESSAGE DICTIONARY
UGZ0005P GUEST TAIL-END CLEAN UP PROGRAM/SCREEN SEND
UGZ0006P OPTIONAL MAP/DEMAP MODULE
UGZ0007P GUEST INTERNAL FUNCTION DISPATCHER
UGZ0008P GUEST HOTKEY(2ND SESSION) CONTROLLER
UGZ0009P GUEST EXTERNAL FUNCTION SWITCHER
UGZ0010P GUEST MENU PROGRAM
UGZ0011P GUEST LU6.2 PROTOCOL PROGRAM
UGZ0012P GUEST-SIDE FUDGE FOR ALAR-INIT INTERNAL DISPATCH
UGZ0013P GUEST HOST= GUEST PROTOCOL MODULE
UGZ0014P GUEST SITE DEFINITION MODULE
UGZ0015P GUEST SIDE SYSTEM STARTUP PROGRAM
UGZ0016P PROGRAM TO INITIALIZE VSAM SWAP FILE
UGZ0017P RESIDENT DYNAMIC TABLE MODULE
UGZ0018P GUEST STATIC TABLE MODULE (towns, states, insurance companies and their locations)
UGZ0019P GUEST STATIC SUBURB TABLE
UGZ0020P GUEST TABLE TRANSMISSION CONTROL MODULE
UGZ0022P GUEST REFERENCE LIST MANAGER
UGZ0023P GUEST TRACE MODULE
UGZ0024P GUEST SIDE NON-RESIDENT SERVICES
UGZ0025P GUEST SWITCH TO ALAR
UGZ0026P GUEST INITIALIZE ALAR BRIDGE
UGZ0028P GUEST-SIDE REPORT QUEUEING INTERFACE PROGRAM
UGZ0029P UMS GUEST PRINTER-TABLE DEFINITIONS
UGZ0030P GUEST-SIDE PRINT TASK
UGZ0032P DISPLAY MODULE/FUNCTION VERSION
UGZ0034P UMS MODULE/FUNCTION STATUS SCREEN
UGZ0034Z PRINTER OVER-RIDE TABLE
UGZ0035P SCREEN PRINT EDIT TABLE
UGZ0036P GUEST-SIDE SCREEN PRINT TASK (UGSP)
UGZ0037P GUEST-SIDE SCREEN PRINT DIRECTOR
UGZ0038P 2ND LEVEL HOST CALL ROUTINE
UGZ0039P CICS BR14 FOR 2ND LEVEL HOST CALL ROUTIND
UGZ0040P GUEST-SIDE LXTABLE POPUP CONTROL
UGZ0060P GUEST-SIDE HOSTIO CONTROL MODULE
HOST-SIDE PROGRAMS

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UHZ0001P</td>
<td>HOST-SIDE FUNCTION DISPATCHER</td>
</tr>
<tr>
<td>UHZ0002P</td>
<td>HOST PROCESS CONTROL TABLE</td>
</tr>
<tr>
<td>UHZ0003P</td>
<td>HOST UTILITIES PROGRAM</td>
</tr>
<tr>
<td>UHZ0004P</td>
<td>HOST TABLE TRANSMISSION MODULE</td>
</tr>
<tr>
<td>UHZ0005P</td>
<td>HOST TOWN-CODE TABLE</td>
</tr>
<tr>
<td>UHZ0006P</td>
<td>HOST PLATE-EDITOR TABLE, OLD FORM</td>
</tr>
<tr>
<td>UHZ0007P</td>
<td>HOST MISC. VARIABLES MODULE (CODE/TEXT)</td>
</tr>
<tr>
<td>UHZ0008P</td>
<td>HOST SECURITY INITIALIZATION PROGRAM</td>
</tr>
<tr>
<td>UHZ0009P</td>
<td>HOST USER DATA STASH ROUTINE</td>
</tr>
<tr>
<td>UHZ0010P</td>
<td>HOST PLATE-EDITOR TABLE ONLY FOR SECTION 5 PLATES</td>
</tr>
<tr>
<td>UHZ0011P</td>
<td>HOST LU6.2 PROTOCOL MODULE</td>
</tr>
<tr>
<td>UHZ0012P</td>
<td>HOST PLATE – EDITOR TABLE ONLY FOR GENERIC PLATES, NEW FORM</td>
</tr>
<tr>
<td>UHZ0013P</td>
<td>HOST HOST=GUEST PROTOCOL MODULE</td>
</tr>
<tr>
<td>UHZ0015P</td>
<td>HOST INITIALIZATION PROGRAM (UGTL/UHTL)</td>
</tr>
<tr>
<td>UHZ0018P</td>
<td>HOST PASSWORD CHANGE CONTROL</td>
</tr>
<tr>
<td>UHZ0019P</td>
<td>HOST-SIDE INTERFACE TO LXTABLE (UGZ0006P)</td>
</tr>
<tr>
<td>UHZ0020P</td>
<td>HOST MESSAGE OUTPUT MODULE</td>
</tr>
<tr>
<td>UHZ0021P</td>
<td>HOST-SIDE MONITOR PROGRAM</td>
</tr>
<tr>
<td>UHZ0022P</td>
<td>HOST-SIDE ALAR CALLED LOOKASIDE PURGE</td>
</tr>
<tr>
<td>UHZ0025P</td>
<td>HOSTSIDE LU6.2 PROTOCOL MODULE FOR THROTTLE REGION</td>
</tr>
<tr>
<td>UHZ0026P</td>
<td>LU6.2 THROTTLE MANAGEMENT TABLE</td>
</tr>
<tr>
<td>UHZ0027P</td>
<td>CJIS DATA TRACE ENABLE/DISABLE</td>
</tr>
<tr>
<td>UHZ0028P</td>
<td>HOST-SIDE CROSS-REGION THROTTLE MODE MODULE</td>
</tr>
<tr>
<td>UHZ0029P</td>
<td>HOST-SIDE TERMINAL NMBER RETURN FOR BATCH #</td>
</tr>
<tr>
<td>UHZ0033P</td>
<td>HOST-SIDE LU6.2 LONG DATA FORM PROTOCOL CALL MODULE</td>
</tr>
<tr>
<td>UHZ0060P</td>
<td>HOST-SIDE HOSTIO MANAGER</td>
</tr>
<tr>
<td>UHZ0061P</td>
<td>HOSTIO TABLE DEFINITIONS</td>
</tr>
<tr>
<td>UHZ0101P</td>
<td>SECURITY RECORD MANAGEMENT</td>
</tr>
<tr>
<td>UHZ0102P</td>
<td>VTAB TABLE LOAD ROUTINE</td>
</tr>
<tr>
<td>UHZ0102S</td>
<td>STATIC VTAB LOAD SUBROUTINE</td>
</tr>
<tr>
<td>UHZ0103P</td>
<td>CORT TABLE LOAD ROUTINE</td>
</tr>
<tr>
<td>UHZ0103S</td>
<td>STATIC CORT LOAD SUBROUTINE</td>
</tr>
<tr>
<td>UHZ0104P</td>
<td>TOWN TABLE LOAD ROUTINE</td>
</tr>
<tr>
<td>UHZ0105P</td>
<td>PASSWORD CHANGER</td>
</tr>
<tr>
<td>UHZ0107P</td>
<td>RETURN TERMINAL FOR CLERK-ID</td>
</tr>
<tr>
<td>UHZ0108P</td>
<td>BUILD TERMID MAP FOR CONVERSIONG</td>
</tr>
<tr>
<td>UHZ1013P</td>
<td>VIRTUAL GUEST HOST CALLER</td>
</tr>
</tbody>
</table>
Appendix F:

Guest Common Area

COBOL Example

* UNINSURED MOTORIST. COMMON-AREA, GUEST SIDE.
* USE AS-IS FOR VS-COBOL. COPY REPLACING INDEX BY POINTER
  FOR COBOL2
05 UGCOMMON-SYSTEM-PROTOCOL-DATA PIC X(64).
  05 FILLER REDEFINES UGCOMMON-SYSTEM-PROTOCOL-DATA.
  10 UGCOMMON-DUPLICATE-WORD-TEMP.
    15 UGCOMMON-DUPLICATE-WORD1 PIC S9(09) COMP.
    15 FILLER REDEFINES UGCOMMON-DUPLICATE-WORD1.
    20 UGCOMMON-DUPLICATE-WORD1-PNTR USAGE INDEX.
    15 UGCOMMON-DUPLICATE-WORD2 PIC S9(09) COMP.
    15 FILLER REDEFINES UGCOMMON-DUPLICATE-WORD2.
    20 UGCOMMON-DUPLICATE-WORD2-PNTR USAGE INDEX.
  10 UGCOMMON-XCTL-LENGTH PIC S9(04) COMP.
  10 UGCOMMON-RESPONSE-LENGTH PIC S9(04) COMP.
  10 UGCOMMON-CURRENT-FUNCTION PIC X(04).
  10 UGCOMMON-ALLON-FLAGS PIC X.
  10 UGCOMMON-ALLON-FLAGS-2 PIC X.
  10 UGCOMMON-PREVIOUS-FUNCTION PIC X(04).
  10 UGCOMMON-RECEIVED-MAP-LENGTH PIC S9(04) COMP.
  10 UGCOMMON-CUR-FUNC-CTRL-FLAGS.
    15 UGCOMMON-PROCESS-BYTE-1 PIC X.
    15 UGCOMMON-PROCESS-BYTE-2 PIC X.
    15 UGCOMMON-PROCESS-BYTE-3 PIC X.
    15 UGCOMMON-PROCESS-BYTE-4 PIC X.
    15 UGCOMMON-USER-FLAGS.
    20 UGCOMMON-F4-STRING PIC X(04).
    20 UGCOMMON-F9-STRING PIC X(04).
  10 UGCOMMON-PREV-MAP-NAME PIC X(07).
  10 UGCOMMON-MESSAGE-TEXT PIC X(01).
    88 UGCOMMON-LAST-MAP-HAD-TEXT VALUE 'Y'.
  10 UGCOMMON-REQUESTED-TRANID PIC X(04).
  10 UGCOMMON-GENERAL-STATUS-FLAG PIC X.
  10 FILLER PIC X(15).
  05 UGCOMMON-MODULE-NAME-AREA PIC X(192).
  05 FILLER REDEFINES UGCOMMON-MODULE-NAME-AREA.
  10 UGCOMMON-PROTOCOL-PROCESSOR PIC X(08).
  10 UGCOMMON-APPLICATION-MODULE-1 PIC X(08).
  10 UGCOMMON-APPLICATION-MODULE-2 PIC X(08).
  10 UGCOMMON-APPLICATION-MODULE-3 PIC X(08).
10 UGCOMMON-CLEANUP-PROCESSOR PIC X(08).
10 UGCOMMON-RETURN-AFTER-HOST PIC X(08).
10 UGCOMMON-MAP-TRANSLATION-TBL PIC X(08).
10 UGCOMMON-SCREEN-SENDBACK-MOD PIC X(08).
10 UGCOMMON-INTERNAL-DISPATCH-MOD PIC X(08).
10 UGCOMMON-INTERNAL-REFERENC-MOD PIC X(08).
10 UGCOMMON-NONRES-SERVICES-MOD PIC X(08).
10 UGCOMMON-LXTABLE-PROCESSOR PIC X(08).
10 UGCOMMON-REPORT-PRINT-PROC PIC X(08).
10 UGCOMMON-LEVEL2-HOST-CALL PIC X(08).
10 UGCOMMON-CICS-BR14-EQUIV PIC X(08).
10 UGCOMMON-HOST-IO-HANDLER PIC X(08).
10 UGCOMMON-PRINTER-INTERFACE PIC X(08).
10 FILLER PIC X(56).
05 UGCOMMON-ASSIST-ADDRESS-AREA PIC X(192).
05 FILLER REDEFINES UGCOMMON-ASSIST-ADDRESS-AREA.
10 UGCOMMON-DATE-Routine-ADDRESS PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-DATE-Routine-ADDRESS.
15 UGCOMMON-DATE-Routine-PNTR USAGE INDEX.
10 UGCOMMON-DATA-NAME-ADDRESS PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-DATA-NAME-ADDRESS.
15 UGCOMMON-DATA-NAME-PNTR USAGE INDEX.
10 UGCOMMON-FROM-HOST-TEXT-ADDR PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-FROM-HOST-TEXT-ADDR.
15 UGCOMMON-FROM-HOST-TEXT-PNTR USAGE INDEX.
10 UGCOMMON-TO-HOST-TEXT-ADDRESS PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-TO-HOST-TEXT-ADDRESS.
15 UGCOMMON-TO-HOST-TEXT-PNTR USAGE INDEX.
10 UGCOMMON-GUEST-CONTROL-TABLE PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-GUEST-CONTROL-TABLE.
15 UGCOMMON-GUEST-CONTROL-PNTR USAGE INDEX.
10 UGCOMMON-MISC-TRANSLATE-TABLES PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-MISC-TRANSLATE-TABLES.
15 UGCOMMON-MISC-TRANSLATE-PNTR USAGE INDEX.
10 UGCOMMON-GENERAL-EDIT-SERVICES PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-GENERAL-EDIT-SERVICES.
15 UGCOMMON-GENERAL-EDIT-PNTR USAGE INDEX.
10 UGCOMMON-SCHELL-SORT-ADDRESS PIC S9(08) COMP.
10 FILLER REDEFINES UGCOMMON-SCHELL-SORT-ADDRESS.
15 UGCOMMON-SCHELL-SORT-PNTR USAGE INDEX.
10 FILLER PIC X(160).
05 UGCOMMON-SECURITY-SAVE-DATA PIC X(24).
05 FILLER REDEFINES UGCOMMON-SECURITY-SAVE-DATA.
10 UGCOMMON-SITE-NAME PIC X(04).
10 UGCOMMON-USER-ID PIC X(04).
10 UGCOMMON-PASSWORD PIC X(04).
10 UGCOMMON-TERMINAL-NAME PIC X(04).
10 FILLER PIC X(08).
05 UGCOMMON-SCROLL-BASE-DATA PIC X(742).
05 FILLER REDEFINES UGCOMMON-SCROLL-BASE-DATA.
10 UGCOMMON-LAST-SCROLL-FUNC PIC X(04).
10 FILLER PIC X(02).
10 UGCOMMON-SCROLL-FORMAT PIC X(12).
10 FILLER REDEFINES UGCOMMON-SCROLL-FORMAT.
15 UGCOMMON-SCROLL-DATA-LINE1 PIC S9(4) COMP.
15 UGCOMMON-SCROLL-SURR-TYPE-TBL PIC X(08).
*                  NONE VALUE LOW-VALUES OR SPACES.
*                  VEHR VALUE '0'.
*                  VEHC VALUE '1'.
*                  VEHT VALUE '2'.
*                  VMOD VALUE '3'.
*                  PERS VALUE '4'.
*                  NOWN VALUE '5'.
*                  PLCY VALUE '6'.
*                  OACD VALUE '7'.
*                  OVIO VALUE '8'.
*                  OWTH VALUE '9'.
*                  CITA/IVIO VALUE 'A'.
*                  ACTN VALUE 'C'.
*                  DECI VALUE 'D'.
*                  PINQ VALUE 'E'.
*                  AUTI VALUE 'F'.
*                  SUBM VALUE 'G'.
*                  SDIP XCLU VALUE 'H'.
*                  REIN EXPA VALUE 'M'.
*                  MSCO VALUE 'V'.
*                  DISP VALUE 'W'.
10 UGCOMMON-SCROLL-KEY    PIC X(67).
10 UGCOMMON-CORE-STATUS-FLAG    PIC X(01).
10 UGCOMMON-PAGE-AREA     PIC X(96).
10 UGCOMMON-PAGE-TABLE REDEFINES UGCOMMON-PAGE-AREA OCCURS 12 TIMES PIC X(08).
10 FILLER REDEFINES UGCOMMON-PAGE-AREA.
15 UGCOMMON-TOP-1       PIC X(08).
15 UGCOMMON-BOTTOM-11   PIC X(88).
10 FILLER REDEFINES UGCOMMON-PAGE-AREA.
15 UGCOMMON-TOP-11      PIC X(88).
15 UGCOMMON-BOTTOM-1    PIC X(08).
10 UGCOMMON-SCROLL-TABLE PIC X(560).
10 FILLER REDEFINES UGCOMMON-SCROLL-TABLE.
15 UGCOMMON-SURR-NUMB OCCURS 140 TIMES INDEXED BY UGC-SURR-PNTR PIC S9(09) COMP.
05 UGCOMMON-APPLICATION-WORK-AREA PIC X(512).
05 FILLER REDEFINES UGCOMMON-APPLICATION-WORK-AREA.
10 UGCOMMON-STANDARD-PARM-AREA PIC X(22).
10 FILLER REDEFINES UGCOMMON-STANDARD-PARM-AREA.
15 UGCOMMON-TEXT-MOVE-FLAG-1 PIC X.
   88 UGCOMMON-LEN-IS-ADDRESS-MODE VALUE 'A'.
   88 UGCOMMON-LEN-IS-FULL-WORD VALUE 'L'.
15 UGCOMMON-INCRIMENT-FLAG PIC X.
   88 UGCOMMON-INCRIMENT-OCUR VALUE 'Y'.
   88 UGCOMMON-DONT-INCRIMENT-OCUR VALUE 'N'.
15 UGCOMMON-DATE-AREA.
   20 UGCOMMON-DATE-INPUT-FORMAT PIC X.
* 0 = BINARY (DO NOT RETURN DAY OF WEEK)
* 1 = GREGORIAN (DO NOT RETURN DAY OF WEEK)
* 2 = JULIAN  (DO NOT RETURN DAY OF WEEK)
* 4 = BINARY (RETURN DAY OF WEEK)
* 5 = GREGORIAN (RETURN DAY OF WEEK)
* 6 = JULIAN (RETURN DAY OF WEEK)

20 UGCOMMON-DATE-RETURN-STATUS PIC X.

* 0 = OK
* 1 = RANGE ERROR
* 2 = NON-NUMERIC INPUT
* 3 = INVALID MONTH
* 4 = INVALID DAY OF MONTH
* 5 = INVALID YEAR
* 6 = INVALID DAY OF YEAR
* 7 = NULL DATE INPUT
* 8 = FORMAT UNKNOWN

20 UGCOMMON-DATE-BINARY-FORMAT.
25 UGCOMMON-DATE-BINARY-FORMAT-N4 PIC S9(04) COMP.
20 FILLER REDEFINES UGCOMMON-DATE-BINARY-FORMAT.
25 UGCOMMON-DATE-JULIAN-FORMAT PIC 9(07).
20 FILLER REDEFINES UGCOMMON-DATE-JULIAN-FORMAT.
25 UGCOMMON-DATE-JULIAN-YEAR PIC 9(04).
25 UGCOMMON-DATE-JULIAN-DAY PIC 9(03).
20 UGCOMMON-DATE-GREG-FORMAT-X.
25 UGCOMMON-DATE-GREG-FORMAT PIC 9(08).
20 FILLER REDEFINES UGCOMMON-DATE-GREG-FORMAT-X.
25 UGCOMMON-DATE-GREG-YEAR PIC 9(04).
25 UGCOMMON-DATE-GREG-MONTH PIC 9(02).
25 UGCOMMON-DATE-GREG-DAY PIC 9(02).
20 UGCOMMON-DATE-DAY-OF-WEEK PIC X(16).
10 UGCOMMON-INVOCATION-ENTRY-INFO PIC X(16).
10 FILLER REDEFINES UGCOMMON-INVOCATION-ENTRY-INFO.
15 UGCOMMON-SCREEN-NAME PIC X(07).
15 UGCOMMON-ENTRY-REASON PIC S9(04) COMP.
88 UGCOMMON-ENTRY-FIRST-TIME VALUE +00.
88 UGCOMMON-ENTRY-ENTER VALUE +01.
88 UGCOMMON-ENTRY-F7 VALUE +02.
88 UGCOMMON-ENTRY-F8 VALUE +03.
88 UGCOMMON-ENTRY-F4 VALUE +04.
88 UGCOMMON-ENTRY-RESCROLL VALUE +05.
88 UGCOMMON-ENTRY-FROM-HOST VALUE +06.
88 UGCOMMON-ENTRY-AFTER-F9 VALUE +07.
88 UGCOMMON-ENTRY-AFTER-F12 VALUE +08.
88 UGCOMMON-ENTRY-AFTER-FUNC VALUE +09.
88 UGCOMMON-ENTRY-AFTER-DUPKEY VALUE +10.
88 UGCOMMON-ENTRY-F9-NOSELECT VALUE +11.
88 UGCOMMON-ENTRY-F5 VALUE +12.
15 UGCOMMON-F4-SURROGATE-TYPE PIC X(01).
88 UGCOMMON-F4-VEHR-TYP-0 VALUE '0'.
88 UGCOMMON-F4-VEHC-TYP-1 VALUE '1'.
88 UGCOMMON-F4-VEHT-TYP-2 VALUE '2'.
88 UGCOMMON-F4-VMOD-TYP-3 VALUE '3'.
88 UGCOMMON-F4-PERS-TYP-4 VALUE '4'.
88 UGCOMMON-F4-NONW-TYP-5 VALUE '5'.
88 UGCOMMON-F4-PLCY-TYP-6 VALUE '6'.
15 UGCOMMON-F4-SURROGATE PIC S9(09) COMP.
15 UGCOMMON-CURRENT-BINARY-DATE.
20 UGCOMMON-CURRENT-BINARY-DATE9 PIC S9(04) COMP.
10 UGCOMMON-RESERVED-SURROGATES.
15 UGCOMMON-RESERVED-SURR-LIST OCCURS 12 TIMES INDEXED BY UGCOMMON-RESERVED-SURR-INDX.
20 UGCOMMON-RESERVED-SURR-TYPE PIC X.
20 UGCOMMON-RESERVED-SURR-VALUE PIC S9(9) COMP.
10 UGCOMMON-CURRENT-MAP-LENGTH PIC S9(09) COMP.
10 UGCOMMON-INPUT-MAP-ADDRESS PIC S9(09) COMP.
10 FILLER REDEFINES UGCOMMON-INPUT-MAP-ADDRESS.
15 UGCOMMON-INPUT-MAP-PNTR USAGE INDEX.
10 UGCOMMON-OUTPUT-MAP-ADDRESS PIC S9(09) COMP.
10 FILLER REDEFINES UGCOMMON-OUTPUT-MAP-ADDRESS.
15 UGCOMMON-OUTPUT-MAP-PNTR USAGE INDEX.
10 UGCOMMON-RESERVED-SURR-ADDR PIC S9(09) COMP.
10 FILLER REDEFINES UGCOMMON-RESERVED-SURR-ADDR.
15 UGCOMMON-RESERVED-SURR-PNTR USAGE INDEX.
10 UGCOMMON-GETMAIN-RESERVED PIC X(16).
* DEFINED THE LAYOUT OF THE GETMAIN RESERVED AREA
10 FILLER REDEFINES UGCOMMON-GETMAIN-RESERVED.
* CONTAINS GETMAIN'd AREA FOR RECORDS FOR VIRTUAL GUEST
* WHEN VIRTUAL GUEST IS RUNNING.
15 UGCOMMON-INTERFACE-ERR-FLAG PIC X.
* SET THIS TO A "Y" IF YOU DO NOT WANT THE ERROR TO COME BACK TO
* YOUR PROGRAM.  ERROR MESSAGES LESS THAN 200000000 NEVER RETURN.
10 UGCOMMON-INTERFACE-DISPATCH-AREA.
15 UGCOMMON-INTERFACE-DISPATCH-TYPE PIC X(01).
  88 UGCOMMON-GUEST-APPL-REQUEST VALUE '0'.
  88 UGCOMMON-HOST-DUPKEY-REQUEST VALUE '1'.
  88 UGCOMMON-VGUEST-APPL-REQUEST VALUE '2'.
  88 UGCOMMON-VHOST-DUPKEY-REQUEST VALUE '3'.
15 UGCOMMON-INTERFACE-DISPATCH-NAME PIC X(04).
15 UGCOMMON-INTERFACE-DISPATCH-SAVE PIC X(24).
10 UGCOMMON-EDITOR-AREA.
15 UGCOMMON-EDITOR-FUNCTION PIC S9(4) COMP.
15 UGCOMMON-EDITOR-LENGTH PIC S9(4) COMP.
15 UGCOMMON-3BYTE-DATE PIC S9(04) COMP.
10 FILLER PIC X(345).
05 UGCOMMON-ERROR-DATA.
 10 UGCOMMON-CURRENT-ERROR-CODE SYNC PIC S9(09) COMP.
10 UGCOMMON-ERROR-TEXT.
15 FILLER PIC X(50).
10 UGCOMMON-3270-CURSOR-POSITION PIC S9(04) COMP.
05 UGCOMMON-PROGRAM-COMMAREA PIC X(2290).
ASSEMBLER Example

* UNINSURED MOTORIST. COMMON-AREA, GUEST SIDE.
UGCOMM chevy
UGCPROTO DS XL64        SYSTEM PROTOCOL DATA
       ORG UGCPROTO
* WITH TWO EXCEPTIONS, THE PROTOCOL AREA IS RESERVED FOR THE CONTROL
* SOFTWARE. APPLICATIONS MAY USE THE DOUBLE-WORD TEMP AND SHOULD NOTE
* THE QUALIFICATION ABOUT PREVIOUS MAP-NAME
UGCDTEMP DS D          GENERAL PURPOSE DOUBLE-WORD TEMPORARY
UGCCLENG DS H          LENGTH FOR USE WITH XCTL.
UGCRDLEN DS H          INQUIRY/REPLY DETAIL LENGTH
UGCCURFN DS CL4        CURRENT FUNCTION-NAME
UGCALLON DS XL1        FLAG BYTE WITH ALL BITS INITIALLY ON
UGCMAPFI EQU B'00000001'  MAP/DEMAP OPERATION IS DEMAP
UGMAPFO EQU B'00000010'  MAP/DEMAP OPERATION IS MAP
UGCMAPFI EQU B'01000000'  ERROR FLAGGING REQUESTED FOR TZA
UGFLG2 DS CL1          ADDITIONAL FLAG BYTE SC1192
UGCF2_SSEND1 EQU B'00000001'   SET BY 1P ON TERMINATING 2NDRY 92
UGCF2_SSEND2 EQU B'00000010'   SET BY 1P ON READMAP IF EN D1 SC1192
* BOTH ABOVE BITS CLEARED IF SSEND2 ON AT MAP READ SC1192
UGCF2_OVRIDE EQU B'00000100'  SET BY 6P FOR OVERRIDE SECURITY SC1193
* CLEARED BY 1P ON FUNCTION CHANGE  SC1193
UGCF2_OVRIDE EQU B'00000100'  SET BY 6P FOR OVERRIDE SECURITY SC1193
UGCPNFLN DS CL4        PREVIOUS FUNCTION-NAME
UGCCRCVML DS AL2       MAP-LENGTH FROM RECEIVE-MAP
UGCCURFL DS CL12       CURRENT FUNCTION CONTROL-FLAGS
UGCPFLG1 EQU UGCCURFL+0,1  PROCESS FLAG BYTE-1
UGCPFLG2 EQU UGCCURFL+1,1  PROCESS FLAG BYTE-2
UGCPFLG3 EQU UGCCURFL+2,1  PROCESS FLAG BYTE-3
UGCPFLG4 EQU UGCCURFL+3,1  PROCESS FLAG BYTE-4
UGCPFLGU EQU UGCCURFL+4,8  USER FLAGS IF NOT (F4+F9)
UGCPFF4S EQU UGCCURFL+4,4  F4 STRING IF (F4)
UGCPFF9S EQU UGCCURFL+8,4  F9 STRING IF (F9)
* THIS INFORMATION COMES FROM THE PROCESS-CONTROL TABLE AND INDICATES
* SUCH INFORMATION AS (1) WHAT FUNCTION KEY FUNCTIONS THE CURRENT FUNCTION
* ACCEPTS, (2) WHAT SURROGATES ARE REQUIRED FOR A F4 ENTRY, ETC.
RMVMV.RMV.UMSMACR3(UGZCOMMA) - 01.11       Columns 00001 00072
UGPREVMA DS CL7        PREVIOUSLY SHOWN MAP-NAME
* NOTE: IF THE USER-PROGRAM IS REUSING THE LAST MAP, BUT HAS DONE
* SOMETHING WHICH MAKES A DATA-ONLY SEND INAPPROPRIATE, IT SHOULD ZAP
* THE PREVIOUS MAP NAME
UGMFGTXT DS CL1        SET BY OUTPUT-PROCESSOR TO C'Y' IF THE
* LAST MAP SENT HAD TEXT IN THE MESSAGE FIELD.
UGTRANID DS CL4        REQUESTED DISPATCH TRANID
UGSTAFLG DS CL1        GENERAL STATUS-FLAG
UGSTSRQ EQU B'00000001'   SIGNON IS REQUIRED
UGSTSALW EQU B'00000010'   SIGNON IS ALLOWED
UGSTFRQ EQU B'00000010'   FUNCTION-SELECTION IS REQUIRED
UGSTLIVE EQU B'10000000'  AREA IS LIVE COMMON-AREA (DOCUMENTARY)
UGSTLOGT EQU B'01000000'   ACTIVE TRANSACTION IS LOGON TRANSACTION  
UGSTHOTK EQU B'00100000'   ACTIVE TRANSACTION IS SECONDARY MODE  
UGST2LIM EQU B'00010000'   SECONDARY MODE IS LIMITED SESSION  
UGSTREFL EQU B'00001000'   REFERENCE LIST ESTABLISHMENT BIT  
* NOTE: THE DIFFERENCE BETWEEN SIGNON REQUIRED AND ALLOWED OCCURS WHEN  
* THE SIGNON SCREEN IS REENTERED BY A CLEAR KEY AS OPPOSED TO ENTERED  
* AS A RESULT OF CICS SIGNON  
*  
ORG  
UGCMODNA DS XL192           MODULE NAME AREA  
ORG   UGCMODNA  
UGCOUTPR DS CL8             OUTPUT-SIDE PROTOCALL PROCESSOR  
UGCMODX DS 0CL(4*8)         MODULES FROM PCTE ENTRY  
UGCMOD1 DS CL8              1ST APPLICATION PROCESSOR MODULE  
UGCMOD2 DS CL8              2ND APPLICATION PROCESSOR MODULE  
UGCMOD3 DS CL8              3RD APPLICATION PROCESSOR MODULE  
UGCCLEAN DS CL8             TAIL-END CLEANUP PROCESSOR  
UGCRTDRN DS CL8             MODULE FOR RETURN AFTER HOST INTERFACE  
UGMAPINO DS CL8             OPTIONAL MAP TRANSLATION TABLE NAME  
UGSENDGBK DS CL8            NAME OF SCREEN SEND-BACK MODULE  
UGDISRTN DS CL8             NAME OF INTERNAL FUNCTION DISPATCHER  
UGREFRTN DS CL8             NAME OF REFERENCE-LIST MANAGEMENT RTN  
UGNRSERV DS CL8             NON-RESIDENT SERVICES MODULE SC1289  
UGLXTPRC DS CL8             LXTABLE PROCESSOR SC0491  
UGRPTPRT DS CL8             REPORT PRINT PROCESSOR SC0591  
UGHOSTL2 DS CL8             LEVEL 2 HOST CALL INTERFACE SC0297  
UGCIBR14 DS CL8             CICS BR14 EQUIVALENT SC0297  
UGHOSTIO DS CL8             HOST IO HANDLER SC0298  
UGSPIMOD DS CL8             STD PRINT INTERFACE SC1200  
ORG  
UGCASRTDS DS XL192          ASSIST ROUTINE/TABLE ADDRESS AREA  
* EFFECTIVLY REDUCED FROM 192 BYTES (48 EPTS) TO 184 BYTES SC1193  
* (46 EPTS) BY OVERRIDE/SUPERVISOR SECURITY MODIFICATIONS SC1193  
ORG   UGCASSRT  
UGCAMOD1 DS 0XL(4*8)        LENGTH OF VECTOR IN UGZ0003P  
UGCADTCR DS A               ADDRESS OF DATE CONVERSION ROUTINE  
UGCADNRN DS A               ADDRESS OF DATA-NAMES TO ADDRESS ROUTINE  
UGCAITIR DS A               ADDRESS OF FROM-HOST TEXT FETCH ROUTINE  
UGCAOTIR DS A               ADDRESS OF TO-HOST TEXT FETCH ROUTINE  
UGCACTLT DS A               ADDRESS OF GUEST CONTROL-TABLE  
UGCAMSTR DS A               ADDRESS OF MISC TRANSLATE TABLES  
UGCEDRTN DS A               ADDRESS OF GUEST CONTROL-TABLE  
UGCASSRTDS A               ADDRESS OF SCHELL-SORT ROUTINE SC0190  
ORG  
UGCFPSEC DS XL24            SECURITY-SAVE DATA  
ORG   UGCFPSEC-8            OVERRIDE PASSWORD AREA SC1193  
UG_OVPASSWD DS CL8          OVERRIDE PASSWORD SC1193  
ORG   UGCFPSEC  
UFHSIT DS CL4               GUEST SITE NAME  
UFHUGUSER DS CL4            GUEST USER-ID  
UFHGISWD DS CL4             GUEST PASSWORD, PART-1  
UFHGETRM DS CL4             GUEST TERMINAL-NAME  
UFHGPSW2 DS CL4             GUEST PASSWORD, PART-2  
UG_OVUSER DS CL4            GUEST OVERRIDE USER ID SC1193  
*  
**
*** W A R N I N G
** FROM THE APPLICATIONS PERSPECTIVE, ALL DATA PRECEEDING THIS
* POINT IS READ-ONLY

ORG
UFGSCROL DS 742CL1 SCROLL BASE DATA
UFGSCROS EQU *-UFGSCROL LENGTH OF DATA
ORG UFGSCROL
UFGSCLFN DS CL4 LAST SCROLL-FUNCTION NAME
DS CL2 UNUSED AT THIS TIME
UFGSCEPL DS 0CL12 FORMAT DATA FOR SCROLL-TABLE
UFGSEPL DS ZL1 #ENTRIES PER LINE
UFGSCPE DS ZL1 #SURROGATES PER ENTRY
UFGSCLN1 DS H FIRST LINE USED ON SCREEN
UFGSCFMT DS 0CL12 FORMAT DATA FOR SCROLL-TABLE
* '0'=VEHR, '1'=VEHC, '2'=VEHT, '3'=VMOD, '4'=PERS, '5'=nown
* '6'=PLCY,
* '7'=OACD, '8'=OVIO, '9'=OWTH, 'A'=CITA, 'C'=ACTN, 'D'=DECI,
* 'E'=PINQ, 'F'=AUTI, 'G'=SUBM, 'H'=SDIP XCLU, 'M'=EXPA, CP1197
* 'V'=MSCO W=DISP
* (25-1STLINE)*(#SURR/ENT)*#ENT/LINE .LE. 140
UFGSKEY DS CL67 SCROLL KEY
UFGSFLAG DS CL1 CORE STATUS FLAG
UFGSPTBL DS 12CL8 PAGE TABLE
UFGSCTBL DS CL560 SCROLL/BROWSE TABLE
ORG
*NOTE: THE CONCEPT HERE IS THAT THE USER IS SUPPOSED TO BE ABLE TO DO A
* F4 SELECT FROM A SCROLL-SCREEN AND VIEW A RECORD. WHEN THE USER THEN
* DECIDES HE HAS NOT SELECTED THE CORRECT RECORD, ENTERING THE SCROLL
* COMMAND AGAIN IS SUPPOSED TO GET HIM THE SAME SCROLL SCREEN. THIS IS
* INTENDED TO BE PROVIDED AT TWO LEVELS: (1) UNCONDITIONAL ABILITY TO
* COME BACK TO THE LAST SCROLL PAGE WITH NO ABILITY TO PAGE-BACK, AND
* (2) CONDITIONAL ABILITY TO COME BACK TO THE LAST SCROLL PAGE AND PAGE
* BACK. THE CONDITIONAL ABILITY IS BASED ON IF ANOTHER APPLICATON HAS
* CORRUPTED THE SCROLL BASE-DATA. BASICALLY, A SCROLL APPLICATION WILL
* PUT HIGH-VALUES IN UFGSFLAG INDICATING IT HAS VALID CONTENT IN THE
* CORE-TABLES. ANY OTHER APPLICATION MAY CORRUPT THIS AREA IF IT DES-
* IRES, BUT MUST FIRST PUT LOW-VALUES IN UFGSFLAG. THE OTHER SCROLL AREA
* ARE NOT TO BE USED EXCEPT BY SCROLL APPLICATIONS. FOR 3270 GUESTS, A
* FUNCTION KEY WILL BE DESIGNATED FOR SCROLL-REENTRY.
* THE CORRUPTABLE AREA BEGINS WITH THE SECOND ENTRY IN UFGSPTBL.
*
UGCSWKA DS 512CL1 DATA-SERVICES/APPL WORK AREA
ORG UGCDSWKA
UGDSTPR DS 0XL22 STANDARD PARM AREA
* THE TEXT-MOVE ROUTINE IS CALLED PASSING COMMON-AREA,SOURCE,LENGTH
* SUBJECT TO THE FOLLOWING FLAGS. THE LENGTH ARGUMENT MAYBE A FULL-
* WORD CONTAINING THE ACTUAL LENGTH, OR THE ADDRESS OF A BYTE FOLLO-
* WING THE LAST TO MOVE. THE USER SPECIFIES IF THE OCCURRENCE COUNT
* IS TO BE INCRCEMTED, THE FIELDS ARE INITIALIZED TO 'AN' AND RESET
* TO THIS VALUE AFTER EACH CALL.
UGCIVF1 DS CL1 TEXT MOVE FLAG-1
* 'A'="LENGTH" PARAMETER IS ADDRESS-MODE
* 'L'="LENGTH" PARAMETER IS FULL-WORD
UGCIVF2 DS CL1 OCCURENCE INCRIMENT FLAG
* 'Y'=INCREMNT OCCURENCE COUNTER
* 'N'=DO NOT INCREMNT OCCURENCE COUNTER

Appendix F – Guest Common Area

UGCDCONA DS 0CL20       DATE CONVERSION AREA
UGCDCFMT DS CL1         INPUT FORMAT
UGCDCSTA DS CL1         RETURN STATUS
UGCDCBIN DS XL2         BINARY FORMAT
UGCDCJUL DS CL7         JULIAN YYYYDDD FORMAT
UGCDCGRE DS CL8         GREGORIAN YYYYMMDD FORMAT
UGCDCDAY DS CL1         DAY OF WEEK#
*  
UGEKTRYI DS 0CL16       INVOCATION ENTRY INFORMATION
UGSCREEN DS CL7         NAME OF SCREEN ON TERMINAL
UGERCODE DS XL2         ENTRY REASON
  * 00=ENTRY FROM CONTROL-SUPERVISOR (FIRST TIME)
  * 01=REENTRY FROM CONTROL-SUPERVISOR AFTER RECEIVE-MAP/ENTER
  * 02=REENTRY FROM CONTROL-SUPERVISOR AFTER RECEIVE-MAP/F7
  * 03=REENTRY FROM CONTROL-SUPERVISOR AFTER RECEIVE-MAP/F8
  * 04=ENTRY FROM CONTROL-SUPERVISOR AFTER RECEIVE-MAP/F4
  * 05=ENTRY FROM CONTROL-SUPERVISOR AFTER RECEIVE-MAP/RESCROLL
  * 06=ENTRY/REENTRY FROM HOST INTERFACE
  * 07=ENTRY AFTER F9 SWITCH
  * 08=ENTRY AFTER F12
  * 09=ENTRY AFTER INTERNAL FUNCTION DISPATCH
  * 10=ENTRY AFTER HOST DUPKEY FUNCTION CHANGE
  * 11=ENTRY WITH CURSOR ABOVE FIRST SCROLL-LINE ON F9 SC0492
  * AND NULL9=YES IN PCTEG ENTRY SC0492
  * 12=F5 (REPRINT) SC1092
UGERSTYP DS CL1         SURROGATE-TYPE FOR F4
UGERSURR DS CL4         SURROGATE FOR F4
UGCDATE DS CL2          DATE, SERIAL FORMAT
  DS 0F
UGSURLST DS 12XL5       RESERVED SURROGATE TYPE/VALUE LIST
UGSURTYE EQU UGSURLST+0,1 SURROGATE TYPE, S/A F4 & F9 TYPES
UGSURVAL EQU UGSURTYP+1,4 SURROGATE VALUE
UGSURLEN EQU *-UGSURLST TABLE LENGTH
*  
UGCLRDTA DS 0XL32       CLEARED BY UGZ0001P ON EACH REENTRY
UGMAPLEN DS F           CURRENT OUTPUT-MAP LENGTH
UGINPMAP DS F           INPUT MAP ADDRESS
UGOUTMAP DS F           OUTPUT MAP ADDRESS
UGINTDTA DS F           HOST INTERFACE DATA AREA
UGMEMLST DS 4F          RESERVED FOR 4 MORE GETMAIN AREAS-SC0590
UGLONGHI EQU UGMEMLST+12,4,C’F’ LONG HOST AREA ADDRESS
* NOTE, ABOVE AREAS DO NOT EXIST IF CELL VALUE IS ZERO
* CONVENTION WILL BE THAT IF THE APPLICATION WILL FREEMAINS UNNEEDED
* AREAS IT WILL CLEAR THE CELL. THE REQUIRED LENGTH FOR THE HOST
* INTERFACE AREA IS NEAR THE START OF THIS AREA.
UGEINTFL DS CL1         HOST INTERFACE ERROR-PROCESSING FLAG.
* APPLICATION MUST SET TO ‘Y’ IF INTERFACE IS TO INTERCEPT HOST DET-
* ECTED ERRORS AND NOT RETURN TO APPLICATION. CLEARED BY INTERFACE.
*  
* DATA ORIENTED TO INTERNAL FUNCTION DISPATCH:
UGDISDTA DS 0CL29       INTERNAL DISPATCH DATA AREA
UGDISPTP DS CL1         INTERNAL DISPATCH TYPE:
  * C’0’=GUEST APPL REQUEST
  * C’1’=HOST DUPKEY REQUEST
  * C’2’=GUEST APPL REQUEST VIRTUAL GUEST
  * C’3’=HOST DUPKEY REQUEST VIRTUAL GUEST
Appendix F – Guest Common Area

**NOTE:**

* THE CONTROL-SUPERVISOR WILL SET THE ERROR-CODE AND TEXT TO BINARY
* ZERO BEFORE EXECUTING THE FIRST PROGRAM. THE HOST PROTOCOL CALL PROG-
* RAM WILL COPY ANY ERROR-CODE FROM THE HOST TO THE ERROR-CODE AREA
* AND RETRIEVE THE TEXT. WHEN THE CONTROL-SUPERVISOR REGAINS CONT-
* ROL (ON THE OUTPUT SIDE) IT WILL RETRIEVE TEXT IF THE ERROR-CODE IS
* NON-ZERO AND THE MESSAGE AREA IS NOT (SPACES OR BINARY-ZEROS)
* THE CONTROL-SUPERVISOR WILL INSURE THAT THE TEXT AREA IS COPIED INTO
* THE MESSAGE AREA FOR THE SCREEN. THE CONTROL-SUPERVISOR WILL SET THE
* THE ALARM BIT IF IT THINKS THAT AN ERROR IS TO BE POSTED (1ST TEXT
* BYTE NON-BLANK). IF IT THINKS THAT THIS AREA ON THE SCREEN IS BLANK
* AND THE INTENT IS TO SEND A BLANK-AREA (1ST 2 BYTES SPACES) IT WILL
* NOT SEND THE FIELD. FOR THESE PURPOSES, SPACES AND BINARY-ZEROS ARE
* TREATED THE SAME IN THE TEXT FIELD.

* THE LENGTH OF THIS AREA IS TAKEN FROM THE PROCESS-CONTROL TABLE
* BUT WILL NOT BE LESS THAN THE VALUE OF THE ABOVE EQUATE AT THE LAST
## Appendix G:

**Functions Descriptions**

<table>
<thead>
<tr>
<th>Page</th>
<th>Screen Name</th>
<th>Screen Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>285</td>
<td>COR</td>
<td>Cash Obligation Resolution Screen</td>
</tr>
<tr>
<td>288</td>
<td>COR1</td>
<td>Cash Obligation Resolution Screen</td>
</tr>
<tr>
<td>290</td>
<td>LH</td>
<td>License History Inquiry</td>
</tr>
<tr>
<td>292</td>
<td>LI</td>
<td>License Inquiry Screen</td>
</tr>
<tr>
<td>294</td>
<td>LN</td>
<td>License Number Scroll</td>
</tr>
<tr>
<td>296</td>
<td>LNO</td>
<td>License Number Scroll</td>
</tr>
<tr>
<td>298</td>
<td>LNS</td>
<td>Social Security Number Scroll</td>
</tr>
<tr>
<td>300</td>
<td>LTH</td>
<td>License Transaction History</td>
</tr>
<tr>
<td>302</td>
<td>MRBS</td>
<td>MRB Sub-menu Screen</td>
</tr>
<tr>
<td>304</td>
<td>NRL</td>
<td>Nonrenew display tickets for license</td>
</tr>
<tr>
<td>306</td>
<td>NRR</td>
<td>Nonrenew display tickets for registration</td>
</tr>
<tr>
<td>308</td>
<td>RIC</td>
<td>Lessee Information</td>
</tr>
<tr>
<td>312</td>
<td>RBS</td>
<td>Registration Bank Scroll</td>
</tr>
<tr>
<td>314</td>
<td>RH</td>
<td>Registration History</td>
</tr>
<tr>
<td>316</td>
<td>RI / URI</td>
<td>Registration / Title Inquiry</td>
</tr>
<tr>
<td>320</td>
<td>RNF</td>
<td>Corporation Scroll by FID</td>
</tr>
<tr>
<td>322</td>
<td>RVN / URVN</td>
<td>NADA Value Guide Inquiry</td>
</tr>
<tr>
<td>325</td>
<td>UMA</td>
<td>Merit Rating Board - At Fault Insurance Claim</td>
</tr>
<tr>
<td>327</td>
<td>UMC</td>
<td>Merit Rating Board - Comprehensive Insurance Claim</td>
</tr>
<tr>
<td>329</td>
<td>UMI9</td>
<td>Merit Rating Board - SDIP Statement Input Data Screen</td>
</tr>
<tr>
<td></td>
<td>UMIQ</td>
<td>Same as UMI9 description</td>
</tr>
<tr>
<td>331</td>
<td>UMO</td>
<td>Merit Rating Summary - Operator Summary</td>
</tr>
<tr>
<td>333</td>
<td>UMON</td>
<td>Merit Rating Summary - Operator Summary</td>
</tr>
<tr>
<td>335</td>
<td>UMVH</td>
<td>Merit Rating Board - Traffic Citation</td>
</tr>
<tr>
<td>337</td>
<td>UMVS</td>
<td>Merit Rating Board - Traffic Citation</td>
</tr>
<tr>
<td>339</td>
<td>UMVI</td>
<td>Merit Rating Board - Traffic Citation</td>
</tr>
<tr>
<td>341</td>
<td>UP, UR, UL</td>
<td>UMS subsystem screen</td>
</tr>
<tr>
<td>243</td>
<td>UPA</td>
<td>Policy Amend Screen</td>
</tr>
<tr>
<td>349</td>
<td>UPH</td>
<td>Policy History screen</td>
</tr>
<tr>
<td>352</td>
<td>UPIC</td>
<td>Policy Information Screen</td>
</tr>
<tr>
<td>355</td>
<td>UPMV</td>
<td>Multiple Vehicle Amend Screen</td>
</tr>
<tr>
<td>359</td>
<td>UPOI</td>
<td>Policy / Operator inquiry screen</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>363</td>
<td>UPTH</td>
<td>Policy History Screen</td>
</tr>
<tr>
<td>366</td>
<td>RN / URN</td>
<td>UMS Registration Scroll</td>
</tr>
<tr>
<td>368</td>
<td>URSN</td>
<td>Reg/VIN owner scroll screen</td>
</tr>
<tr>
<td>370</td>
<td>URSR</td>
<td>Reg scroll by Reg screen</td>
</tr>
<tr>
<td>372</td>
<td>URSV</td>
<td>Reg scroll by VIN Screen</td>
</tr>
<tr>
<td>274</td>
<td>UVH</td>
<td>Vehicle Policy History Screen</td>
</tr>
<tr>
<td>377</td>
<td>VT</td>
<td>Vehicle Title History</td>
</tr>
<tr>
<td>380</td>
<td>ULP</td>
<td>UMS Person Name Scroll</td>
</tr>
<tr>
<td>382</td>
<td>RA</td>
<td>Registration Amend</td>
</tr>
</tbody>
</table>
COR Function

TITLE: CASH OBLIGATION RESOLUTION SUMMARY

DESCRIPTION: The COR function is used to display reg/owner information and the total amount owed for a registration. To use this screen, the plate type and reg number must be entered.

This screen can “hop” to RI or UVH by changing the function code to the desired screen and pressing the F6 key.

COR is modular, written with specific use subprograms, each being a reusable object. To see how to use any of them, see the relevant discussion below.

NOTE: A typical use of the COR screen would be to find the total amount owed against the registration and to look at the details by hopping to COR1.

COMPONENTS: Guest input program
UGS1191P
501191001 Enter Keys
501191002 Invalid Function Key
501191003 No surrogate - F4 F6
501191004 No surrogate F9
501191005 Key change required
501191006 Registration select required
501191007 No F9 surrogate

Guest output program
UGS1192P
501192001 Press F9

Host program(s)
UHS1193P
501193001 Bad link to UHS1194P
501193002 Bad link to UHS1195P
501203014 Bad link to UHS1197P
501193003 Bad link to UHS1198P
501193004 Bad link to UHS1199P
501193005 Bad link to Message Log

UHS1194P
501194001 Bind Error
501194002 Inquiry complete
501194003 Inquiry Failed
501194004 Key not found

UHS1195P
501195001 Bind Error
501195002 Too many actions
501195003 Inquiry Complete
501195004 Inquiry Failed

UHS1196P
501196001 Bind Error
501196002 Too many actions
501196003 Inquiry Complete
501196004 Inquiry Failed

UHS1197P
501197001 Bind Error
501197002 Too many tickets
501197003 Inquiry Complete
501197004 Inquiry Failed

UHS1198P
501198001 Bind Error
501198002 Too many citations
501198003 Inquiry complete
501198004 Inquiry Failed

UHS1199P
501199001 Bind Error
501199002 Too Many MSCOS
501199003 Inquiry Complete
501199003 Inquiry Failed

Lr
UHS1194L
UHS1195L
UHS1198L
UHS1199L

Map
UGS1190M

Lx table
UGS1190T
Copy books
UGZCOMMY    (UGS1191P, UGS1192P)
UHZCOMMY    (UHS1193P)
UIS119AY     (UGS1191P, UGS1192P,
              UHS1193P, UHS1194P)
UHRREGSY     (UHS1193P)
UILXCNST     (UGS1192P)
UHU1LOGY     (UHS1193P, UHS1194P,
              UHS1195P, UHS1197P,
              UHS1198P, UHS1199P)
UHS1194Y     (UHS1193P, UHS1194P)
UHS1195Y     (UHS1193P, UHS1195P)
UHS1196Y     (UHS1196P)
UHS1197Y     (UHS1197P)
UHS1198Y     (UHS1193P, UHS1198P)
UHS1199Y     (UHS1193P, UHS1199P)
UGS120AY     (UGS1191P)
UIR1240Y     (UGS1191P)
MREY1OWK     (UMS1193P)
UHS1197Y     (UHS1193P, UHS1197P)
UHS1196Y     (UHS1194P)

External name
COR

Internal name
COR2 (to access by reg)
COR1, COR3, URSR

AVAILABLE FUNCTION KEYS:

F1   - Main Menu
F2   - UMS menu
F3   - Not available
F4   - Not available
F5   - Not available
F6   - Screen hopping
F7   - Not available
F8   - Not available
F9   - Hop to COR1 for details
F10  - Not available
F11  - Not available
F12  - Not available

NOTE:  F6 requires function change
COR1 Function

TITLE: CASH OBLIGATION RESOLUTION DETAIL

DESCRIPTION: The COR1 function is used to display detail information with the amounts owed and the total amount owed for a registration. To use this screen, first enter COR, then press F9. No function code change is needed.

COR1 is also modular, calling the various subprograms as needed.

This screen can “hop” to RI or UVH by changing the function code to the desired screen and pressing the F6 key.

This screen can “hop” to an appropriate screen, such as the CIC screen, for a citation by changing the function code, putting the cursor on the appropriate line, and hitting enter.

To return to COR, simply press F9 again.

NOTE: A typical use of the COR1 screen would be to display all the items that cause a fee to be owed against the registration.

COMPONENTS:

Guest input program
UGS1201P

Guest output program
UGS1202P

Host program(s)
UHS1120P

Lr
UHS1195L
UHS1196L
UHS1197L
UHS1198L
UHS1199L
UHS1190L

Map
UGS1200M

Lx table
UGS1200T
Copy books
UGZCOMM (UGS1201P, UGS1202P)
UHZCOMM (UHS1203P)
UGS120AY (UGS1201P, UGS1202P)
UGS120BY (UGS1201P, UGS1202P, UHS1203P)
UIS120CY (UHS1203P)
UHRREGS (UHS1203P)
UILXCNST (UGS1202P)
UHUILOG (UHS1195P, UHS1196P, UHS1197P, UHS1198P, UHS1199P, UHS1190P)
UHS1195Y (UHS1203P, UHS1195P)
UHS1196Y (UHS1203P, UHS1196P)
UHS1197Y (UHS1203P, UHS1197P)
UHS1198Y (UHS1203P, UHS1198P)
UHS1199Y (UHS1203P, UHS1199P)
UHS1190Y (UHS1203P, UHS1190P)

External name
COR1

Internal name
COR4 (to access by reg)

AVAILABLE FUNCTION KEYS:

F1 - Main Menu
F2 - UMS subsession
F3 - Not available
F4 - Screen hopping
F5 - Not available
F6 - Screen hopping
F7 - Page backward
F8 - Page forward
F9 - Return to COR
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F6 and F4 require function change
**LH Function**

**TITLE:** LICENSE HISTORY INQUIRY

**DESCRIPTION:** The LH function is used to request the history of a person's name and license number changes. Note that a change in a person's date of birth is considered the same as a change of name. The screen displays the last three name changes as well as the last eight license numbers held by that person. For Commercial Drivers, the LH “Extension Screen” (LHE), which can only be accessed from LH, will display CDL-specific endorsement and restriction data.

**COMPONENTS:**

*Guest input program*
UGL0060P

*Guest output program(s)*
UGL0061P
UGL0062P

*Host program(s)*
UHL0060P
UHL0070P
UHL0080P
UHL0095P

*Map*
UGL0060M
UGL0062M

*Copy books*
UGZCOMMC (UGL0060P, UGL0061P)
LI06 (UGL0061P, UGL0060C)
REGEQU (UGL0062P)
UGZCOMMA (UGL0062P)
UMZCOMMA (UGL0062P)
UGTRTABL (UGL0062P)

*External name*
LH

*Internal name*
LH10 (Person lookup via current and previous Mass license)
LH20 (Person lookup via non-Mass license)
LH30 (Person lookup via pers-surrogate number)
LH40 (Endorsement and restriction lookup via surrogate number for LHE only)

AVAILABLE FUNCTION KEYS:

F1  - End session
F2  - Main Menu
F3  - Not available
F4  - Not available
F5  - Not available
F6  - Jump to screen
F7  - Not available
F8  - Not available
F9  - Not available
F10 - Not available
F11 - Rescroll
F12 - Not available
**LI Function**

**TITLE:** UNIQUE LICENSE INQUIRY SCREEN

**DESCRIPTION:** The LI function is used to request information pertaining to a current or previous driver’s license number. It obtains personal information (name, mailing address and residence) as well as restriction data, previous name, license status, endorsements, and placard information.

**COMPONENTS:**

*Guest input program*

UGL1070P

- 200001004 - Invalid previous indicator enter X
- 200001005 - LI screen displayed please proceed
- 200001016 - Invalid Social Security Number
- 105003001 - Invalid Mass license
- 105006003 - License invalid
- 200001006 - License number not found
- 108000002 - State invalid
- 200001014 - Social security number must be numeric
- 906002012 - Social security number not found
- 213000012 - Enter License Number or SSN

*Guest output program*

UGL1071P

- 200001007 - Inquiry process complete
- 230001002 - Unable to obtain endorsements date. . .try again
- 230001003 - Unable to obtain cell data. . .try again
- 212017129 - Gen. Permit exp date: Request blu field is in error

*Host program(s)*

- LI07 Lic Inquiry by Lic# UHL0070P
- LI08 Lic Inquiry by OOS# UHL0080P
- LI09 Lic Inquiry by SSN UHL0090P
- LI10 Lic Inquiry by Surrogate UHL0100P, UHL0110P
- LI12 Endorsement Inquiry by Surrogate UHL0120P
- LB10 Generalized Fetcher of Brefs UHL0420P
- PR05 Passenger Restriction Inquiry UHL3205P
- LI96 Generate expiration date (only permit for) UHL0960P
- LXP1 PDPS Pers-Ext Type “6” Lookup UHL0610P

*Map*

UGL1070M
Copy books
UGZCOMMC  (UGL1070P, UGL1071P)
UGL1070G   (UGL1070P, UGL1071P)
UMLLI12O   (UGL1071P)
UMSLXZT    (UGL1071P)
UMLLB10O   (UGL1070P, UGL1071P)
UMLLB10R   (UGL1070P, UGL1071P)
UMLLI12R   (UGL1070P, UGL1071P)
UMLLXP1R   (UGL1070P, UGL1071P)
UMLLXP1O   (UGL1070P, UGL1071P)
UMLP0SR    (UGL1070P, UGL1071P)
UMLP0SO    (UGL1070P, UGL1071P)
UMLLI96R   (UGL1070P, UGL1071P)
UMLLI96O   (UGL1070P, UGL1071P)

LX table
UGL1070T

External name
LI

Internal name
LI07 (Person lookup via current and previous Mass license)
LI08 (Person lookup via non-Mass license)
LI09 (Person lookup via social security number)
LI10 (Person lookup via person surrogate number)

AVAILABLE FUNCTION KEYS:

F1  - Main Menu
F2  - UMS Submenu screen
F3  - Not available
F4  - Not available
F5  - Not available
F6  - Jump to screen
F7  - Not available
F8  - Not available
F9  - Not available
F10 - Not available
F11 - Not available
F12 - Not available
LN Function

TITLE: LICENSE NUMBER SCROLL

DESCRIPTION: The LN function is used to display any duplicate license numbers in the system. The license number is a required field. This function is invoked in one of two ways:

♦ By entering LN in the function field
♦ Automatically, when a duplicate license number is detected by any of the other licensing functions.

Once LN is invoked by another function, the user can return to the original function by positioning the cursor at the desired license information and pressing F9. Up to 7 licenses display per page.

COMPONENTS:

Guest input program
UGL0020P
202001004  - First record displayed
202001001  - License number is invalid
202001003  - End of set encountered

Guest output program
UGL0021P
202001003  - End of set encountered

Host program(s)
UHL0020P - License Number Scroll

Map
UGL0020M

Copy books
REGEQU
UGZCOMMA
UHZCOMMA
UGTRTABL (UGL0020P)

External name
LN

Internal name
LI04
AVAILABLE FUNCTION KEYS:

F1 - Main Menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Select from scroll and refresh previous screen
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
LNO Function

TITLE: OUT-OF-STATE LICENSE NUMBER SCROLL

DESCRIPTION: The LNO function displays any duplicate out-of-state license numbers in the system. License number and state are required fields. The LNO function is invoked in one of two ways:

- By entering LNO in the function field
- Automatically, when a duplicate out-of-state license number is detected by any of the other licensing functions.

When LNO is invoked by another function, the user can return to the original function by positioning the cursor at the desired license information and pressing F9. Up to 7 licenses display per page.

COMPONENTS:

Guest input program
UGL0240P
105009001 - Value invalid
202001003 - End of set encountered
202001004 - First record displayed

Guest output program
UGL0241P
203001002 - No records for this key on database

Host program(s)
UHL0520P - LD08 Duplicate OOS Scroll

Map
UGL0240M

LX table
UGL0240T

Copy books
UMSLXMAC
REGEQU
UGZCOMMA
UGL0240C
UHZCOMMA
UGTRTABL (UGL0240P)
External name
LNO

Internal name
LD08

AVAILABLE FUNCTION KEYS:

F1 - Main Menu
F2 - UMS Submenu Screen
F3 - Not available
F4 - Select from scroll Screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Select from scroll and refresh
      previous screen
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
LNS Function

TITLE: SOCIAL SECURITY NUMBER SCROLL

DESCRIPTION: The LNS function displays any duplicate social security numbers in the system. LNS is invoked in one of two ways:

♦ By entering LNS in the function field
♦ Automatically, when a duplicate social security number is detected by any of the other licensing functions.

When LNS is invoked by another function, the user can return to the original function by positioning the cursor at the desired social security number information, and pressing F9. Up to 7 social security numbers display per page.

COMPONENTS:

Guest input program
UGL0220P
202001004 - First record displayed
105009001 - Invalid social security number
202001003 - End of set encountered

Guest output program
UGL0221P
202001003 - End of set encountered

Host program(s)
UHL0500P - LD04 Duplicate SSN Scroll
203001002 - No records for this key on database

Map
UGL0220M

LX table
UGL0220T
200001016 - Invalid Social Security Number

Copy books
UMSLXMAC
REGEQU
UGZCOMMA
UGL0220C
UHZCOMMA
UGTRTABL (UGL0220P)
External Name
LNS

Internal Name
LD04

AVAILABLE FUNCTION KEYS:

F1 - Main Menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Select from scroll
    and refresh previous screen
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
LTH Function

TITLE: LICENSE TRANSACTION HISTORY

DESCRIPTION: LTH provides a list of all transactions, along with transaction history, associated with a license. Each transaction displays the transaction date, associated fee, batch number, and explanation of the transaction code. This function also includes summarized cash records, NDR BREF records, transaction codes and descriptions, in addition to the license holder’s name, address, and date of birth. All changes made on the LC screen are displayed, as well as all changes to date.

The LTH function is invoked by entering LTH in the FUNCTION field. Up to 15 transactions are displayed per page.

COMPONENTS:

Guest input program
UGL0260P
206003001 - Function key or enter invalid at this time
200001016 - Lic# or SS# required
206003008 - F7 invalid beginning of set
206003009 - F8 invalid at end of set
206003010 - Phase count-internal error

Guest output program
UGL0261P
200004001 - Logic internal error
206004002 - No history records for this person

Host program(s)
UHL0070P (LI07)
UHL0080P (LI08)
UHL0100P (LI10)
UHL0110P
UHL0420P (LB10 - BREFS)
208003001 - Bind error
208003002 - Ready error

Map
UGL0260M

LX table
UGL0260T
200001016 - Invalid Security Number

*Transaction code table*
UGL0262P - Bref description table

*Copy books*
UMSLXMAC
REGEQU
UGZCOMMA
UGL0260C
UHZCOMMA

*External name*
LTH

*Internal name*
LI07/LB10 or LI08/LB10

**AVAILABLE FUNCTION KEYS:**

F1 - Main Menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Next function with current key
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available
**MRBS Function**

**TITLE:** MRB DUPLICATE LICENSE SCROLL

**DESCRIPTION:** The MRBS function displays any duplicate license numbers, both MA and out-of-state, in the system. This function differs from the LN function in that it makes no distinction between current and previous license numbers when displaying duplicates. Additionally, if the keywords “NOLICENSE” for license and “XX” for license state are used, the MRBS function will accept a name and date of birth which limits the display to only those “nolicense” person records that match. As an added bonus on NOLICENSE, if the date of birth is null and the name begins with an ‘*’, the display will show a list of all companies with names that match.

The MRBS function can only be invoked by another function for duplicate resolution. The user can then return to the original function by positioning the cursor at the desired person/company information and pressing F9. Up to seven persons/companies are displayed per page.

**COMPONENTS:**
- **Guest input program**
  - UGM0001P

- **Guest output program**
  - UGM0002P

- **Host program(s)**
  - UHM0002P

- **Map**
  - UGM0001M

- **LX Table**
  - UGM0001T

- **Copy books**
  - UMSLXZT
  - UGZCOMMC
  - UGM0001C
  - UHZCOMMY

- **External name**
  - MRBS
Internal name
MRBX

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Select from scroll and refresh previous screen
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
NRL Function

TITLE: NONRENEW DISPLAY TICKETS FOR LICENSE

DESCRIPTION: The NRL function is used to review all of the outstanding tickets and/or excise bills for a license number.

When the license is entered, NRL will display unpaid parking tickets, abandoned vehicle citations, or excise bills marked against the license.

This function displays the license number, state, the license holder’s date of birth, license holder’s last name, first name, middle name, number of items marked against this particular license, the three digit code and name for each town that marked a ticket or bill for the license, the type of outstanding obligation marked (abandoned vehicle citation, excise non-payment or unpaid parking tickets), the year of the outstanding obligation, and the numbers or alphabetic characters assigned to the ticket or bill.

The NRL function is invoked by entering NRL in the FUNCTION field.

COMPONENTS: Guest input program
UGN0040P
220002000 - Enter license
206003010 - Phase count - internal error
206003001 - Function key or enter invalid at this time
220002001 - F7 invalid-beginning of set
220002002 - F8 invalid-end of set

Guest output program
UGN0041P
226004001 - Logic internal - error
220002003 - End of set

Host program (s)
UHL0100P - LI10 - License inquiry by surrogate number
UHL0110P
UHL0070P - LI07 – License inquiry by license number
UHL0080P - LI08 – License inquiry by OOS number
UHN0040P - NR40 – Nonrenew indexed ticket inquiry

Map
UGN0040M
LX table
UGN0040T

Copy books
UMSLXMAC
REGEQU
UGZCOMMA
UGN0040C
UHZCOMMA

External name
NRL

Internal name
NR40

AVAILABLE FUNCTION KEYS:

F1 - Main menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Next function with current key
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available
NRR Function

TITLE: NONRENEW DISPLAY TICKETS FOR REGISTRATION

DESCRIPTION: The NRR function is used to review all of the outstanding tickets and/or excise bills for a registration number.

After a plate type, registration number, and plate color (optional) are entered NRR will display each parking ticket, abandoned vehicle ticket, or excise bill currently marked.

This function displays the current plate color (if it was not entered), manufacturer’s vehicle identification number indicating the vehicle marked for non-renewal, vehicle year, manufacturer’s description of the vehicle, owner’s last name, first name, middle name, number of items marked against a particular registration, the three digit code and name for each town that marked a ticket or bill for the registration, the type of outstanding obligation marked (abandoned vehicle citation, excise non-payment or unpaid parking tickets), the year of the outstanding obligation, and the numbers or alphabetic characters assigned to the ticket or bill.

The NRR function is invoked by entering NRR in the FUNCTION field.

COMPONENTS:

Guest input program
UGN0020P
220001000 – Enter registration
206003010 - Phase count – internal error
206003001 - Function key or enter invalid at this time
220001001 – F7 invalid-beginning of set
220001002 - F8 invalid-end of set

Guest output program
UGN0021P
226004001 - Logic internal - error
220001003 - End of set

Host program (s)
UHR1063P - URI5 – Registration inquiry by reg number
UHR1067P
UHN0040P – NR40 – Nonrenew indexed ticket inquiry
Map
UGN0020M

LX table
UGN0020T

Copy books
UMSLXMAC
REGEQU
UGZCOMMA
UGN0020C
UHZCOMMA

External name
NRR

Internal name
NR40

AVAILABLE FUNCTION KEYS:

F1 - Main menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Next function with current key
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available
R1C Function

TITLE: R1C SCREEN

DESCRIPTION: The R1C screen is used to inquire on, add, change, or delete lessee information associated with a registration. The required key data fields are plate type and registration number. The plate color is optional and, if not entered, will become the default color for the plate type entered. The screen will display the lessee’s license/fid number, name, address, and the lease effective date. The vehicle owner’s name and address will also be displayed, along with the vehicle information.

There are 4 actions that can be processed on this screen. They include “I,” Inquiry; “A,” Add; “C,” Change; and “D,” Delete.

All the actions require a registration to be entered. If there are duplicate records on the database for the registration entered, a screen of duplicate key choices is displayed to allow the user to select the desired record.

In addition to the user going directly to the R1C screen, the R1A/R1B process will pass the user to R1C whenever a tax exempt code of “L” is entered. In this situation, the registration information is passed and the only action allowed is “A.” Once the lessee information is entered, the user is passed back to the R1B screen.

To process a lessee inquiry, an action code of “I” is entered with the required key data fields.

To process a lessee add, an action code of “A” is entered with the required key data fields. Once the vehicle and owner information is retrieved and displayed, the user is required to enter the primary lessee information. If the lessee is an individual, then the license number must be entered. The license state is only needed when the license number is not from Massachusetts. If the license number is not on the database, the user is allowed to add it by supplying the lessee’s name, date of birth, and mailing address. The residence address is optional. If the license is a duplicate on the database, then a duplicate resolution screen will be displayed. Optionally, the user may enter the lessee’s name and date of birth. A screen displaying matching names will be displayed, allowing the user to select the desired record. If the lessee is a corporation, then an FID number must be entered. If the FID is not on the database, the user
will receive an error message that it was not found. If the FID is a
duplicate on the database, then a duplicate resolution screen will be
displayed. A secondary lessee may also be entered. The
secondary lessee can only be an individual. If the license number
entered for the secondary lessee is not found, it will be added using
the same address information as for the primary lessee. For an
individual lessee, the garage code is optional and, if not entered,
will default to the lessee’s mailing city. For a corporation lessee,
the garage code must be entered. Other fields that may be entered
are insurance company number and lease effective date. If the
lease effective date is not entered, it will default to the current date.
The F12 key is used to apply the updates to the database.

To process a lessee change, an action code of “C” is entered with
the required key data fields. When the lessee record has no
secondary lessee and the primary lessee is a corporation, the user
will only be able to add a secondary lessee. When the lessee
record has no secondary lessee and the primary lessee is an
individual, the user can add an FID number (this will make the
individual the secondary lessee) or add a secondary lessee. When
the lessee record has a primary and secondary lessee, the user can
EOF the primary lessee (making the secondary lessee the primary
lessee) or change the secondary lessee. Other fields that may be
changed are garage code, insurance company code, lease effective
date, and lease end date. When the lease end date is entered, the
lease is considered to have ended.

To process a lessee delete, an action code of “D” is entered with
the required key data fields. The system will then require the user
to press F12 twice to delete the record. This action should only be
used to delete a lessee record that was entered in error. The change
action should be used to end a lease.

Inquiry actions are allowed for all users. A user must obtain
authorization to perform add/change actions and delete actions.

The F6 key can be used for screen hopping. The user will be able
to screen hop to screens that have a registration, license, or FID
number as their primary key. For example, the user may F6 to RI
using the displayed registration or F6 to LI using the displayed
lessee’s license number.

The F9 key is used to resolve duplicates from registration, license,
and non-worn scroll screens.
The F12 key will perform database updates. In the case where R1A/R1B transfers the user to R1C, the F12 key will return the user to R1B, and that program will perform the database updates.

**COMPONENTS:**  
*Guest input program*  
UGR2301P  

*Guest output program*  
UGR2302P  

*Host programs*  
UHR2303P, UHR2304P  

*Map*  
UGR2300M  

*LxTable*  
UGR2300T  

*Copy books*  
UGZCOMMY (UGR2301P, UGR2302P)  
UHZCOMMY (UHR2303P)  
UIR230AY (UGR2301P, UGR2302P, UHR2303P)  
UIR230BY (UHR2303P, UHR2304P) UIR219BY (UHR2303P)  
UIR219CY (UGR2302P)  
UIR219DY (UGR2301P)  
UILXCNST (UGR2301P, UGR2302P, UHR2303P)  
UHUILOGY (UHR2303P)  
UIR1OWNY (UHR2303P)  
UIR1510Y (UHR2303P)  
UHR1VISY (UHR2303P)  
UHRREGSY (UHR2303P)  
UHR1ALGY (UHR2304P)  
MREYIOWK (UHR2303P, UHR2304P)  
MREYEDIT (UHR2303P)  
MRMYZIPM (UHR2303P)  
MCCYPOST (UHR2304P)  
MUMYOREG (UHR2304P)  

*External name*  
R1C  

*Internal name*  
R116
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Main Menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Screen Hopping
F7 - Not available
F8 - Not available
F9 - Resolves duplicates from Registration, License and Nown scroll screens
F10 - Not available
F11 - Not available
F12 - Update
**RBS Function**

**TITLE:** REGISTRATION BANK SCROLL

**DESCRIPTION:** The RBS function is used to review a list of lienholder information by lienholder code or name, with the lienholder type as an optional secondary key. Code and name cannot both be entered, and the code must be greater than zero and less than 32768.

**COMPONENTS:**

- **Guest input program**
  UGR2201P
  - 323001001 Enter keys
  - 323001002 Entry not valid
  - 323001003 Enter error
  - 323001004 End of set
  - 323001005 Top of set
  - 323001006 Invalid type
  - 323001007 Can’t have both
  - 323001008 Must have one
  - 323001009 Leading spaces
  - 323001010 Embedded spaces
  - 323001011 Alphanumeric only
  - 323001012 Code too large
  - 323001013 Code not numeric
  - 323001014 Invalid state
  - 323001015 Invalid zip
  - 323001016 Second key invalid

- **Guest output program**
  UGR2202P
  - 323001020 F8 for more
  - 323001021 No more records
  - 323001022 Resolve duplicates

- **Host program**
  UHR2203P
  - 323001025 LR error
  - 323001026 DML error
  - 323001027 No record for key
  - 323001028 Invalid key
  - 323001029 Can’t have both
  - 323001030 Must have one
  - 323001031 Leading space
  - 323001032 Embedded spaces
  - 323001033 Alphanumeric only
Map
UGR2200M

Copy books
UILXCNST (ALL)
UGZCOMMY (UGR2201P, UGR2202P)
UHZCOMMY (UHR2203P)
UGR220AY (UGR2201P, UGR2202P)
UGR220BY (UGR2201P)
UIR220CY (UGR2202P, UHR2203P)
UHUILOGY (UHR2203P)

External name
URBS

Internal name
BS02

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Main Menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Not available
F7 - Page forward
F8 - Page backward
F9 - Select from scroll screen
F10 - Not available
F11 - Not available
F12 - Not available
RH Function

TITLE: REGISTRATION HISTORY

DESCRIPTION: The RH function is used to display a registration’s owner and vehicle information. Both the current vehicle and any previous vehicles that were attached to the registration are displayed. The key fields (fields to be entered) are plate type, registration number, and color. All three fields are required. If color is not entered, a default color is assumed based on the other two fields.

Note: A typical use of this screen would be to find all vehicles that were attached to a specific registration.

Multiple pages may be displayed on this screen by using the standard F7 and F8 for page forward and back.

This screen can be “hopped” from to a UMS screen (eg. RI) or an ALAR screen (eg. TH). To transfer to another screen and carry either the registration or vehicle information, the user can cursor down, select the desired detail line and press F4.

After transferring to a non-scroll screen the rescroll option is available to return to RH by changing the function back to RH and pressing the F11 key.

COMPONENTS:

Guest input program
UGR1251P
326001001 Enter keys
326001002 No surrogate F4
326001003 F7 not valid
326001004 No more records
326001005 Entry not valid

Guest output program
UGR1252P
326002001 F8 for more
326002002 No more records

Host program
UHR1253P
326003000 No key data
326003001 No VEHR on VEHC
326003002 LR error
326003003 No registration found
326003004 Owner missing
326003005 DBKEY error
326003006 Invalid entry key
326003009 Invalid DBKEY
326003010 Bind error
326003011 Bad link registration status
326003012 Vehicle not found, get registration

Map
UGR1250M

Copy books
UILXCNST  (UGR1251P,UGR1252P)
UGZCOMMY  (UGR1251P,UGR1252P)
UHZCOMMY  (UHR1253P)
UGR125AY   (UGR1251P, UGR1252P)
UGR125BY   (UGR1251P)
UIR125CY   (UGR1252P, UHR1253P)
UHUILOGY   (UHR1253P)
UHRREGSY   (UHR1253P)
MREYIOWK   (UHR1253P)

External name
RH

Internal name
RH01

AVAILABLE FUNCTION KEYS:

F1 -  End session
F2 -  UMS Menu
F3 -  Not available
F4 -  Cursor select from scroll screen
F5 -  Not available
F6 -  Not available
F7 -  Page forward
F8 -  Page backward
F9 -  Not available
F10 - Not available
F11 - Rescroll
F12 - Not available
### URI Function

**TITLE:** REGISTRATION/TITLE INQUIRY

**DESCRIPTION:** The URI function is used to inquire on registration and title information. The inquiry is done by entering one or more of the following keys:

- Plate type and registration
- Vehicle identification number (VIN)
- Title number
- License number

At least one of these keys must be entered. The inquiry searches for an exact match on the information entered. If an exact match is not found, an error message displays back to the user indicating no data was found. If a duplicate is encountered for a registration number, processing is transferred to the Reg Scroll by Registration screen (URSR) to resolve the duplicate. If a duplicate is encountered for a VIN, the processing is transferred to the Reg Scroll by VIN screen (URSV). If a duplicate is encountered for a license, processing is transferred to the License Number Scroll screen (LN).

To resolve a duplicate from one of the duplicate resolution screens, the user can move the cursor down to select the desired line by pressing either the F4 or F9 key. There is no difference between the F4 and F9 key except the user needs to change the function code to URI before pressing the F4 key.

**NOTE:** A typical use of the URI function would be to find the registration, title and vehicle information associated with a given key.

**COMPONENTS:**

- Guest input program
  - UGR5061P
  - 311001021 - No surrogate for F4
  - 311001022 - Entry not valid
  - 311001002 - Enter keys
  - 311001001 - Last 6 characters numeric
  - 311001038 - Invalid VIN
Guest output program
UGR5062P
311001003 - Inquiry process complete
311001037 - Error when confirming registration
311001018 - Duplicate registration
311001019 - Duplicate confirm registration

Host program(s)
UHR5063P
311001007 - Entry not found
311001008 - Database error
311001009 - IDMS-Error
311001018 - Duplicate information
311001039 - No duplicate listing
311001040 - Clerk not found
311001041 - CLRK IDMS error
311001042 - CLRK-IO-Error
311001043 - Office type not found
311001044 - Office type IDMS-Error
311001045 - Office type IOError
315063001 - UHRSREGS link error
315063005 - Bind match VEHR
315063010 - Vehr match not found
315063015 - Match VEHR IO error
315063020 - Bind RSLT error
315063030 - RSLT IO error

UHR5064P
311001010 - A record matching entered key was not found
311001011 - Database error
311001012 - IDMS error
311001039 - No duplicate list
311001040 - Clerk not found
311001041 - CLRK IDMS error
311001042 - CLRK IO error
311001043 - Office type not found
311001044 - Office type IDMS error
311001045 - Office type IO error
315063035 - BIND MTCH VEHC
315063040 - MTCH VEHC not found
315063045 - MTCH VEHC Io error
315063050 - BIND RSLT error
315063055 - RSLT not found
315063060 - RSLT IO error
UHR5065P
- 311001013 - Entry not found
- 311001014 - I/O error
- 311001015 - IDMS error
- 311001039 - No duplicate list
- 311001040 - Clerk not found
- 311001041 - CLRK IDMS error
- 311001042 - CLRK I/O error
- 311001043 - Office type not found
- 311001044 - Office type IDMS error
- 311001045 - Office type I/O error
- 315063065 - BIND PERS
- 315063070 - PERS not found
- 315063075 - PERS I/O error
- 315063080 - BIND MTCH VEHR
- 315063085 - MTCH VEHR not found
- 315063090 - MTCH VEHR I/O error
- 315063095 - BIND RSLT error
- 315063100 - RSLT not found
- 315063105 - RSLT I/O error

UHR5066P
- 311001001 - Duplicate title
- 311001004 - Entry not found
- 311001005 - I/O error
- 311001006 - IDMS error
- 315063110 - BIND MTCH VEHC
- 315063115 - MTCH VEHC not found
- 315063120 - MTCH VEHC I/O error
- 315063125 - BIND RSLT error
- 315063130 - RSLT not found
- 315063135 - RSLT I/O error

UHR5067P
- 311001004 - Bad program call
- 311001016 - LR error
- 311001017 - DML error
- 311001013 - Record keys not found
- 311001046 - Inactive revocation

Map
UGR5060M

LX table
UGR5060T
Copy books
UGZCOMMY (ALL)
MREYBRAN (UHR5067P)
MREYRCOD (UHR5067P)
MREYMYSY (UHR5063P)
MREYMYWY (UHR5063P)
UIR5067Y (UHR5067P)
UHUUILOGY (UHR5063P, UHR5064P, UHR5065P, UHR5066P, UHR5067P)
UHRREGSY (UHR5067P)
UHRRTLSY (UHR5067P)
UGRIBLKY (UGR5061P)
MREYIOWR (UHR5063P)

External name
URI

Internal name
URIK, URIL, URIM, URIN

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Transfer key information to selected screen
F7 - Not available
F8 - Not available
F9 - Select from scroll screen
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
**RNF Function**

**TITLE:** CORPORATION SCROLL BY FID

**DESCRIPTION:** The RNF function is used to request and display a list of corporations by Federal ID (FID). The key fields (fields to be entered) are FID and Zip Code. The FID field is the only required field. The Zip Code field is optional, but may be used to narrow the search.

**NOTE:** A typical use of this screen would be to find the corporation if the FID is known.

Multiple pages may be displayed on this screen by using the standard F7 and F8 for page forward and back.

This screen can be “hopped” to from a UMS screen (eg. R1A) or an ALAR screen (eg. OP1, RNM or RP) when a duplicate FID is encountered. To transfer control back with the FID information, the user can cursor down, select the desired FID and press F9.

**COMPONENTS:**

*Guest input program*
UGR1231P
- 322001001 - Invalid entry
- 322001002 - Enter keys
- 322001003 - Enter error
- 322001004 - Enter FID
- 322001005 - Zip not numeric
- 322001006 - Beginning of set
- 322001007 - End offset
- 322001008 - Zip 1 required
- 322001009 - Invalid FID

*Guest output program*
UGR1232P
- 322001070 - No more records
- 322001071 - F8 for more

*Host program*
UHR1233P
- 322001041 - Bad access indicator
- 322001042 - No matching records
- 322001043 - NOWN not found for surrogate
- 322001044 - DBKEY not found
- 322001045 - LR-error
322001046 - DML error
322001047 - Database error
322001048 - NOWN not found for surrogate
322001049 - LR logic error
322001051 - Bad first time indicator

Map
UGR1230M

Copy books
UILXCNST (UGR1231P)
UGZCOMMY (UGR1231P, UGR1232P)
UHZCOMMY (UHR1233P)
UGR123AY (UGR1231P, UGR1232P)
UGR123BY (UGR1231P)
UIR123CY (UGR1232P, UHR1233P)
UHULOGY (UHR1233P)

External name
RNF

Internal name
UR09

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Not available
F7 - Page forward
F8 - Page backward
F9 - Select from scroll screen
F10 - Not available
F11 - Rescroll
F12 - Not available
URVN Function

TITLE: National Automobile Dealers Association (NADA) Inquiry

DESCRIPTION: The URVN function is used to display NADA information related to a vehicle identification number (VIN). The user must enter the VIN to use this screen. However, the model year and the make fields are required if the model year of the vehicle is prior to 1981.

The URVN screen allows the user to inquire on the NADA Valu-Guide, giving the vehicle’s weight, MSRP, loan value, retail value, trade-in value, series, submake, and style.

This screen can be “hopped” to from URI/UVH using F6, from UPA using F3, or from URSR/URSV/URSN with an F4 cursor select.

When the user has been transferred to the URVN screen by pressing the F4 or F9 key on a scroll screen (e.g. URSR screen), the user may return to the scroll screen in order to make a new selection. This is done by changing the function code to the appropriate scroll screen’s function and pressing the F11 key.

When the user has been transferred to the URVN screen by pressing the F3 key on a cursor select screen (e.g. UPA screen), the user may return to the cursor select screen in order to make a new selection by simply pressing the F11 key.

When a tape of a new UMS release is sent to insurance companies, they have 60 days to make changes to the guest software and maps to make the applications more suitable to their individual business needs. For this reason, it is necessary to maintain two versions of the software. During these 60 days, the old version of URVN may be accessed by entering the function code “TRVN,” while the function code “URVN” will execute the recently modified modules. After the 60-day period, TRVN will no longer be available.

NOTE: A typical use of the URVN screen would be to find NADA information related to a VIN.

COMPONENTS: Guest input program
UGR2141P
314001001 - VIN Required
314001002 - Year not numeric
314001003 - Make required
314001004 - Year required
314001005 - Entry not valid
314001006 - Enter keys
314001007 - Bad surrogate value
314001008 - Bad surrogate value
314001009 - Odom required

Guest output program
UGR2142P
314001015 - Inquiry complete

Host program
UHR2143P
314001020 - VIN not found
314001021 - DB error
314001022 - Temporary storage not found
314001023 - Temporary storage write error
314001024 - NADA - program not found
314001025 - Link problem
314001026 - Temporary storage delete error
314001027 - Temporary storage item error
314001028 - LR error
314001029 - Temporary storage read error
314001055 - Bad system RC
314001056 - Vehr record not found
314001057 - Vehc record not found

LR
UHR1143L

Map
UGR2140M

Lx Table
UGR2140T

Copy books
UGZCOMMY (UGR1141P,UGR1142P)
UHZCOMMY (UHR1143P)
UGR214AY (UGR1141P,UGR1142P)
UGR214BY (UGR1141P)
UIR214CY (UGR1142P,UHR1143P)
UGZ1RFLY (UGR1142P)
UHUILOGY (UHR1143P)
External name
URVN

Internal names
UR10

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Limited secondary session (requires function change)
F4 - Select from scroll screen (requires function change)
F5 - Not available
F6 - Screen hopping (requires function change)
F7 - Not Available
F8 - Not Available
F9 - Not available
F10 - Not available
F11 - Return to primary session
F12 - Not available
**UMA Function**

**TITLE:** MERIT RATING BOARD - AT FAULT INSURANCE CLAIM

**DESCRIPTION:** The UMA function displays detail information about an at-fault insurance claim and allows the addition of new claims or the update of existing ones.

**COMPONENTS:**
- *Guest input program*
  UGM0030P

- *Guest output program*
  UGM0031P

- *Host program(s)*
  UHM0001P
  UHM0030P

- *Map*
  UGM0030M

- *LX Table*
  UGM0030T

- *Copy books*
  UHMCTOWN
  UHMCLAIM
  UMSLXZT
  UGZCOMMCC
  UGM0030C
  UHZCOMMCC
  UHZCOMMY
  MMRYNTRT

- *External name*
  UMA

- *Internal name*
  MRB0 (Person lookup via license, state or name and date of birth for NOLICENSE/XX)
  UMA1 (Retrieve/update at-fault claim information)
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Screen hop using person surrogate
F7 - Not available
F8 - Obtain next subordinate claim
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Add new or update existing claim

NOTE: F6 requires a function change


**UMC Function**

**TITLE:** MERIT RATING BOARD - COMPREHENSIVE INSURANCE CLAIM

**DESCRIPTION:** The UMC function displays detail information about a comprehensive insurance claim and allows the addition of new comprehensive claims or the update of existing ones.

**COMPONENTS:**
- Guest input program
  UGM0040P
- Guest output program
  UGM0041P
- Host program(s)
  UHM0001P
  UHM0040P
- Map
  UGM0040M
- LX Table
  UGM0040T
- Copy books
  UHMCTOWN
  UHMCLAIM
  UMSLXZT
  UGZCOMMC
  UGM0040C
  UHZCOMMC
  UHZCOMMY
- External name
  UMC
- Internal name
  MRB0 (Person lookup via license, state or name and date of birth for NOLICENSE/XX)
  UMC1 (Retrieve comprehensive claim information)
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Screen hop using person surrogate
F7 - Not available
F8 - Obtain next subordinate claim
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Add new or update existing comprehensive claim

NOTE: F6 requires a function change
UMI9 Function

TITLE: MERIT RATING BOARD - OPERATOR INQUIRY

DESCRIPTION: The UMI9 function requests Safe Driver Insurance Plan (SDIP) statement information on one to ten operators per operator inquiry.

COMPONENTS:

Guest programs
UGM0020P
UGM0021P
UGM0029P

Host programs
UHM0001P
UHM0022P

Maps
UGM0020M
UGM0021M

LX Table
UGM0020T
UGM0021T

Copy books
MMRYCTWN
MMRYS_DIP
MMRYS_DIP2
MMRY455D
MMRY455P
UGM0020C
UGM0021C
UGZCOMMC
UHMMINQT
UHZCOMMC
UIR1COMR
UMSLXZT

External name
UMI9 (Accept Operator Inquiry Requests)

Internal name
MRB0 (Person lookup via MMRV-PERS surrogate number)
UMIQ (Inquiry display)
UMI3  (Operator incident retrieval)

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Resolve duplicate drivers license (MRBS)
F10 - Not available
F11 - Not available
F12 - Not available
**UMO Function**

**TITLE:** MERIT RATING BOARD - OPERATOR SUMMARY

**DESCRIPTION:** The UMO function displays a scrollable list of summary information about incidents associated with a person or company. Incidents displayed are citations, at-fault accidents, comprehensive claims, and insurance inquiries.

A request type option allows the display to be limited to only one of the above incident types (which will include reversed incidents and violation detail information for citations), a combination of all the above types (which will not include reversed incidents), or a list of only surchargeable incidents.

Additionally, a separate list can be displayed showing all MRB administrative requests which have been processed for the selected person.

**COMPONENTS:**

*Guest input program*
UGM0010P

*Guest output program*
UGM0011P

*Host program(s)*
UHM0001P
UHM0010P

*Map*
UGM0010M

*LX Table*
UGM0010T

*Copy books*
UHMCTOWN
UMSLXZT
UGZCOMMC
UGM0010C
UHZCOMMC
UHZCOMMY
UHM0060C
External name
UMO

Internal name
MRB0 (Person lookup via license, state or name and date of birth for NOLICENSE/XX)
UMO1 (Retrieve incident/administrative request information for person or company)

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
UMON Function

TITLE: MERIT RATING BOARD - OPERATOR SUMMARY

DESCRIPTION: The UMON function displays a scrollable list of summary information about citation incidents associated with a company.

COMPONENTS:

- Guest input program
  UGM0070P

- Guest output program
  UGM0071P

- Host program(s)
  UHM0001P
  UHM0010P

- Map
  UGM0070M

- LX Table
  UGM0070T

- Copy books
  UHMCTOWN
  UMSLXZT
  UGZCOMMC
  UGM0070C
  UHZCOMMC
  UHZCOMMY

- External name
  UMON

- Internal name
  MRB0 (Company lookup via Company Name and/or vehicle registration.)
  UMO1 (Retrieve incident/administrative request information for company)
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
UMVH Function

TITLE: MERIT RATING BOARD - TRAFFIC CITATION

DESCRIPTION: The UMVH function displays history information about a traffic citation.

COMPONENTS:

- Guest input program
  UGM0052P

- Guest output program
  UGM0053P

- Host program(s)
  UHM0050P

- Map
  UGM0052M

- LX Table
  UGM0052T

- Copy books
  UHMCTOWN
  UMSLXZT
  UGZCOMMC
  UGM0052C
  UHZCOMMY

- External name
  UMVI

- Internal name
  UMV1 (Retrieve citation information)

AVAILABLE FUNCTION KEYS:

- F1 - End session
- F2 - Main menu
- F3 - Not available
- F4 - Not available
- F5 - Not available
- F6 - Screen hop using person/company surrogate
- F7 - Not available
- F8 - Not available
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F6 requires a function change
**UMVS Function**

**TITLE:** MRB DUPLICATE CITATION SCROLL

**DESCRIPTION:** The UMVS function displays any duplicate citations in the system. License number and license state are optional fields that can be used to restrict the display of duplicate citations to specific operators.

The UMVS function can only be invoked by another function for duplicate resolution. The user can then return to the original function by positioning the cursor at the desired citation information and pressing F9. Up to five citations are displayed per page.

**COMPONENTS:**

- Guest input program
  UGM0055P

- Guest output program
  UGM0056P

- Host program(s)
  UHM0055P

- Map
  UGM0055M

- LX Table
  UGM0055T

- Copy books
  UMSLXZT
  UGZCOMMC
  UGM0055C
  UHZCOMMY

- External name
  UMVS

- Internal name
  UMVX
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Select from scroll and refresh previous screen
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F4 requires a function change
UMVI Function

TITLE: MERIT RATING BOARD - TRAFFIC CITATION

DESCRIPTION: The UMVI function displays detail information about a traffic citation.

COMPONENTS:

Guest input program
UGM0050P

Guest output program
UGM0051P

Host program(s)
UHM0050P

Map
UGM0050M

LX Table
UGM0050T

Copy books
UHMCTOWN
UMSLXZT
UGZCOMMC
UGM0050C
UHZCOMMY

External name
UMVI

Internal name
UMV1 (Retrieve citation information)

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - Main menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Screen hop using person/company surrogate
F7 - Not available
F8 - Not available
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F6 requires a function change
**UP, UR, UL Function**

**TITLE:** INSURANCE SUBMENU (UMS SUBMENU SCREEN)

**DESCRIPTION:** The UP function is an insurance function of the Uninsured Motorist System (UMS). The insurance submenu screen displays seven (7) functions:

- **UPA** – Policy Amend
- **UPH** – Policy Holder History Inquiry
- **UPTH** – Policy Transaction History Inquiry
- **UPIC** – Policy Information
- **UPOI** – Policy Operator Inquiry
- **UVH** – Vehicle History
- **UPMV** – Multiple Vehicle Amend

Each of these functions could be invoked by entering an appropriate function code in the Function field.

The UP function is invoked by entering UP in the FUNCTION field.

The UR Registration Submenu of the Uninsured Motorist System (UMS) displays eleven (11) functions:

- **R1A** - RMV-1 APPLICATION
- **RA** - REGISTRATION AMEND
- **RI** - REGISTRATION/TITLE INQUIRY
- **RH** - REGISTRATION HISTORY
- **RBS** - BANK/LIENHOLDER SCROLL
- **URSN** - REGISTRATION SCROLL/NAMES
- **URSR** - REGISTRATION SCROLL/REG
- **URSV** - REGISTRATION SCROLL/VIN
- **URN** - CORPORATE-OWNER DATA SCROLL
- **RNF** - CORPORATE-OWNER FID SCROLL
- **URVN** - NADA INQUIRY

Each of these functions can be invoked by entering an appropriate function code in the Function field.

The UR function is invoked by entering UR in the FUNCTION field.

The UL Licensing Submenu of the Uninsured Motorist System (UMS) displays seven (7) functions:...
LI - LICENSE NUMBER INQUIRY
LN - LICENSE NUMBER SCROLL
LNO - OUT-OF-STATE NUMBER SCROLL
LNS - SOCIAL SECURITY NUMBER SCROLL
ULP - PERSON-NAME SCROLL
LH - LICENSE HISTORY
LTH - LICENSE TRANSACTION HISTORY

Each of these functions can be invoked by entering an appropriate function code in the Function field.

The UR function is invoked by entering UR in the FUNCTION field.

COMPONENTS:

Guest Side Menu program
UGZ0010P
103003001 - Submenu in PCTEG absent from module

Copy books
REGEQU
UGZCOMMA
UGZGCTLT

External name
UP, UR, UL

Internal name
UP, UR, UL

AVAILABLE FUNCTION KEYS:

F1 - Main menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Not available
F7 - Not available
F8 - Not available
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available
UPA Function

TITLE: POLICY AMEND

DESCRIPTION: The UPA screen is used to inquire, bind, cancel, or reinstate policies. It will also clear an unpaid premium and amend the policyholder or the vehicles on an insurance policy. The key data fields are insurance company code, policy number, policy effective date, and policy type. The policy type field will either be “C” (Commercial) or “P” (Private). The screen will display policy and policyholder information along with all the vehicles, registrations, and owners currently insured on the policy.

To process a policy inquiry, an action code of “I” is entered with the policy key data fields. If no action code is entered, it will default to “I.”

To process a policy cancellation, an action code of “C” is entered with the policy key data fields. Three types of policy cancellations exist: cancellation prior to the policy becoming active or bound, advance cancellation of an active policy (in which the cancellation effective date is in the future), or immediate cancellation of an active policy (in which the cancellation effective date is the current date or sooner). If the cancellation is of an advance policy (policy not active or bound), the effective date must equal the policy effective date. If the cancellation is an advance cancel of an active policy, the effective date must be greater than the current date and within the policy effective dates. If the cancellation is an immediate cancel of an active/expired policy, the effective date must be less than or equal to the current date and within the policy effective dates. A reason code must be entered to complete the cancellation. Unpaid premiums are entered when an amount exists. The F12 key is used to apply updates to the database. After the cancellation is complete, the policy status and status date will reflect the cancellation. If an advanced cancellation was performed, a message stating the policy has been advance-cancelled is displayed on the message line.

To process a policy reinstatement, an action code of “R” is entered with the policy key data fields. The cancellation effective date and reason code, along with any unpaid premiums, are displayed. When reinstating cancelled policies, the reinstatement effective date defaults to the policy’s status date (equal to the cancellation effective date). When reinstating advance-cancelled policies, the reinstatement effective date defaults to the cancellation effective date and may not be changed. A reinstatement reason code must
be entered to complete the reinstatement. If the reinstatement reason code entered is 'BOA' (reinstatement Board of Appeals), then the user must have CAB authority. The reinstatement source field defaults to the user's ID. Additionally, a reinstatement will erase the full amount of the unpaid premium. The PF12 key is used to apply updates to the database. After the reinstatement is complete, the policy status will return to active and the cancellation/reinstatement effective dates and reason codes on the screen are cleared out. Also, for reinstatements of cancelled policies, the policy status date will reflect the reinstatement effective date.

To process a policy bind, an action code of “B” is entered with the policy key data fields. The policy must be in bound status to process a bind action. The policy number must contain a “#” in the first digit and be followed by fifteen (15) numeric characters. If the bound policy is found, a new insurance company policy number must be entered and it must not contain a “#” in the first digit. Policy effective date and expire date may also be changed. The F12 key is used to apply updates to the database.

To process a policy clear unpaid premium, an action code of “U” is entered with the policy key data fields. The policy must be in cancelled status and the unpaid premium amount must be greater than zero to process a clear unpaid premium action. A reason code must be entered to complete the clear unpaid premium action. The F12 key is used to apply updates to the database. After the clear unpaid premium action is complete, the policy unpaid premium amount will reflect a zero balance.

To process a policy policyholder amend, an action code of “P” is entered with the policy key data fields. The policy must not be in cancelled status to process a policyholder amend action. A license code and license state must be entered to complete the policyholder amend action. If the license state is MA, the license must be on the database. If the license is not MA, a new out-of-state license will only be added if there currently is no policyholder on the policy. If an out-of-state operator is being added, last name, DOB, mail address, city, state, and zip must be entered. If the policyholder has no valid license, “NOLICENSE” must be entered for the license number and “XX” must be entered for the license state. If a duplicate license number/state is entered, a screen of duplicate key choices is displayed to allow the user to select the desired duplicate. The F12 key is used to apply updates to the database. After the policyholder amend is complete, the policyholder information will reflect the new policyholder data.
Security will only allow insurance companies to update their own policies. Inquiries may be performed on policies for any company.

The F3 key provides access to a limited secondary session. The Registration/Title Inquiry screen (URI) will be used in this secondary session example. After the user has displayed the information for a policy on the UPA screen, which includes the VINS and registrations connected with that policy, he/she changes the function code from UPA to URI, places the cursor on the desired VIN/Registration detail line and presses F3. This will transfer him to the URI screen and display the information for the selected registration. To return to the UPA screen, he/she may press the Clear key or the F1 key. Once he/she has returned to the UPA screen, a message is displayed saying that the primary session has been resumed.

F6 can be used for screen hopping. The user will be able to screen hop to screens that have a policy or license number as their primary key. For example the user may F6 to UPTH using the policy displayed on UPA, or F6 to UPH using the policyholder’s license displayed on UPA.

When a tape of a new UMS release is sent to the insurance companies, the companies have 60 days to make changes to the guest software and maps to make the applications more suitable to their individual business needs. For this reason, it is necessary to maintain two versions of the software. During these 60 days, the UPCR screen, which UPA is replacing, may be accessed by entering the function code “TPCR,” while the function code “UPA” will execute the recently modified modules. After the 60-day period, TPCR will no longer be available.

COMPONENTS:

Guest input program

UGU3121P

410001001 - PLEASE ENTER KEYS
410001002 - POLICY EFFECTIVE DATE MISSING
410001003 - POLICY # MISSING OR INVALID
410001004 - INS COMPANY # INVALID
410001005 - REASON CODE INVALID
410001006 - ENTRY NOT VALID
410001008 - POLICY # CONTAINS INVALID CHARACTERS
410001010 - NO SURROGATE FOR F9
410001011 - REQUIRED FIELD MISSING, PLEASE KEY
410001012 - CANCELLATION EFFECTIVE DATE
INVALID
410001013 - REINSTATEMENT EFFECTIVE DATE INVALID
410001014 - NO ERRORS - PRESS F12 TO UPDATE
410001024 - NO SURROGATE FOR F4
410001025 - F7/F8 NOT ALLOWED
410001035 - POLICY IN FUTURE CANCEL STATUS ALREADY
410001038 - MUST HAVE CAB AUTHORITY TO USE REASON CODE OF BOA
410001058 - CANT F12 AFTER UPDATE COMPLETE
410001060 - MUST CHANGE TO A NEW PLCY HLDR TO F12
410001101 - UNKNOWN NOWN NOT ALLOWED
410001102 - COMM EFF DATE INVALID
410001107 - 24 VEHC WARNING
410001105 - INV. POLICY TYPE

Guest output program
UGU3122P
410001014 - NO ERRORS - PRESS F12 TO UPDATE
410001038 - MUST HAVE CAB AUTHORITY TO USE REASON CODE OF BOA
410001060 - MUST CHANGE TO A NEW PLCY HLDR TO F12

Host programs
UHU3123P
410001015 - LR ERROR
410001016 - IDMS ERROR
410001017 - RECORD MATCHING KEYS NOT FOUND
410001018 - INQUIRY COMPLETE
410001019 - INQUIRY COMPLETE-PLCY CANT BE CANCELLED
410001020 - INQUIRY COMPLETE-PLCY CANT BE REINSTATED
410001022 - INQUIRY COMPLETE - PROCEED WITH CANCEL
410001023 - INQUIRY COMPLETE- PROCEED WITH REINSTATE
410001026 - BAD PROGRAM CALL
410001027 - INQUIRY COMPLETE - NOT AUTHORIZED TO PDATE
410001048 - LICENSE NOT FOUND
410001049 - OOS LICENSE NOT FOUND - PROCEED WITH ADD
410001050 - OOS LICENSE NOT FOUND - ADD NOT ALLOWED
410001051 - INQUIRY COMPLETE - UNPAID PREM CANT BE CLEARED
410001052 - INQUIRY COMPLETE - BIND REQUEST IS NOT VALID
410001053 - INQUIRY COMPLETE - PLCY CANT BE AMENDED
410001054 - INQUIRY COMPLETE - BIND ERROR, PLCY ALREADY EXISTS
410001055 - INQUIRY COMPLETE - PROCEED WITH CLEAR
410001056 - INQUIRY COMPLETE - PROCEED WITH BIND
410001057 - INQUIRY COMPLETE - PROCEED WITH AMEND
410001101 - UNKNOWN NOWN NOT ALLOWED
410001103 - COMPANY NOT FOUND 1
410001104 - COMPANY NOT FOUND 2
431230001 - BAD LINK LOGGING MOD

**UHU3124P**

410001015 - LR ERROR
410001016 - IDMS ERROR
410001028 - DATA HAS CHANGED SINCE UPDATE STARTED, RECHECK
410001029 - UPDATE COMPLETE
410001030 - VEHICLE REGISTRATION NOT FOUND
410001031 - UPDATE FAILED
410001032 - POLICY NOT FOUND
410001037 - UPDATE COMPLETE - PLCY HAS BEEN ADVANCE CANCELLED
410001066 - POLICY EFFECTIVE DATE INVALID
410001067 - POLICY EFFECTIVE DATE > 8000
410001068 - INVALID CANCELLED DATE
410001070 - PREMIUM NOT NUMERIC
410001071 - INVALID CANCELLED REASON
410001072 - REINSTATE DATE NOT EQUAL TO POLICY DATE
410001073 - REINSTATE EFFECTIVE DATE > 8000
410001074 - INVALID ZIP
410001075 - INVALID REINSTATE REASON
410001076 - UNPAID REASON CODE IS CLEAR
410001077 - PERSON SURROGATE EQUAL ZERO
410001078 - INVALID ADDRESS
410001079 - INVALID NAME
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Limited secondary session (requires function change)
F4 - Screen hopping required function change
F5 - Not available
F6 - Screen hopping (requires function change)
F7 - Not available
F8 - Not available
F9 - Not Available
F10 - Not available
F11 - Not available
F12 - Update
UPH Function

TITLE: POLICY HISTORY INQUIRY

DESCRIPTION: The UPH function is used to display insurance policy information for a policyholder’s license number or FID number. The user must enter the license number to use this screen. The state code must also be entered for out-of-state licenses. If more than one person has the same license number/state, the user is transferred to one of two license number scroll screens to resolve the duplicate license number. For Massachusetts duplicate license numbers, the user is transferred to the License Number Scroll (LN) screen. Out-of-state license duplicates are resolved on the Out-Of-State License Number Scroll (LNO) screen.

The F3 key provides access to the UPA and UPTH screens. The UPA (Policy Amend) screen will be used in this secondary session example. After the user has displayed the policy history for a person using the UPH screen, the user places the cursor on the desired policy detail line, changes the function code from UPH to UPA and presses F3. This will transfer the user to the UPA screen and display the information for the desired policy. At this time, the user is in inquiry mode, but may change the action code and press the Enter key to change the mode to update. When the user is finished inquiring or updating a policy, he/she may press the Clear key or the F1 key to return to where he/she was on the UPH screen, thus resuming the primary session.

The F11 key is used for rescrolling. When the user has been transferred to the UPH screen because he/she pressed the F4 or F9 key on a scroll screen, the user may decide he/she has selected the wrong person. The user may return to the scroll screen, in order to make a new selection, by changing the function code to the appropriate scroll screen’s function and pressing the F11 key.

When a tape of a new UMS release is sent to the insurance companies, the companies have 60 days to make changes to the guest software and maps to make the applications more suitable to their individual business needs. For this reason, it is necessary to maintain two versions of the software. During these 60 days, the old version of UPH may be accessed by entering the function code “TPH,” while the function code “UPH” will execute the recently modified modules. After the 60-day period, TPH will no longer be available.
NOTE: A typical use of the UPH screen would be to find the insurance policy information for policyholders using their license number/state.

COMPONENTS:  

**Guest input program**

UGU3041P

- 411001016 - F7 is not valid
- 411001015 - No surrogate found for F4
- 411001012 - Entry not valid
- 411001011 - Required field missing - Please key
- 411001014 - End of set - No more records
- 411001013 - F8 for more records

**Guest output program**

UGU3042P

- 411001006 - No policy records for this person
- 411001001 - F8 for more records
- 411001002 - No more records

**Host program**

UHU3043P

- 411001003 - A record matching entered key(s) was not found
- 411001006 - No policy records for this person
- 411001004 - LR error
- 411001007 - Bad program link
- 411001008 - Database error
- 412043001 - Response not normal
- 412043002 - Response not normal

**Map**

UGU3040M

**Lx table**

UGU3040T

**Copy books**

- UGZCOMMY (UGU2041P, UGU2042P)
- UHZCOMMY (UHU2043P)
- UGU3041Y (UGU2041P, UGU2042P)
- UGU3UPHY (UGU2041P)
- UIU3040Y (UGU2042P, UHU2043P)
- UGZ2RFLY (UGU2042P)
- UHUILOGY (UHU2043P)
- UHU3DIPY (UHU2043P)
External name
UPH

Internal names
UU22, UU23, UU24

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Limited secondary session
F4 - Select from scroll screen
F5 - Not available
F6 - Screen hopping
F7 - Page backward
F8 - Page forward
F9 - Resolve duplicates from License Scroll screens
F10 - Not available
F11 - Rescroll
F12 - Not available

NOTE: F3, F4, and F6 requires function change
**UPIC Function**

**TITLE:** POLICY INFORMATION CHANGE OF CARRIER REQUEST

**DESCRIPTION:** UPIC is used to process policy change of carriers. This function allows the user to enter a new policy with up to 24 vehicles. The user is only allowed access to UPIC after successful inquiry of the operator has been done on the UPOI screen. The policy information, name, and address information from the UPOI screen carries over to the UPIC screen for further processing. The UPIC screen has eight pages, with each page holding up to three vehicles.

The program checks each vehicle entered for a duplicate registration or VIN on the database. If a duplicate exists, the user is transferred to the appropriate scroll screen to resolve the duplicate condition.

**NOTE:** A typical use of the UPIC function would be to change the insurance carrier when only the operator’s license number is known.

**COMPONENTS:**

*Guest input program*  
UGU4021P

- 413001004 - F8 for more or F12 to update
- 413001005 - Record not found
- 413001009 - F7 to page backward or F12 to update
- 413001008 - Duplicates exist - press enter to resolve
- 413001010 - No REG found for vehicle
- 413001006 - LR error
- 413001002 - Update complete
- 413001022 - F12 to update
- 413001012 - Policy number invalid
- 413001013 - Required field missing
- 413001014 - F7/forward or F8/backward
- 413001015 - Please enter required keys
- 413001016 - No more vehicles to display
- 413001017 - Duplicate REG entered
- 413001018 - Enter all vehicle info before F8
- 413001019 - Invalid characters keyed
- 413001020 - Duplicate VIN entered
- 413001021 - Entry not valid
- 413001025 - Need at least one vehicle for F12
- 413001027 - Enter VIN or reg for town code
- 413001028 - New town code entered, F8 or F12 to update
413001029 - New town code entered, F7 or F12 to update
413001026 - New town code entered, press F12 to update
413001045 - Corporate name required
413001046 - Address required
413001051 - Last vehicle
413001055 - Bad DOB
413001056 - Bad DOB convert date
413001057 - Bad current convert date

**Guest output program**

**UGU4022P**

413001018 - Enter all vehicle info before F8
413001016 - No more vehicles
413001014 - F7 or F8
413001028 - New town code entered, F8 or F12 to update
413001129 - New town code entered, F7 or F12
413001004 - F8 or F12 to update
413001009 - F7 or F12 to update
413001008 - Resolve duplicates
413001005 - Not found
413001010 - REG not found
413001006 - LR error
413001002 - Update complete
413001022 - F12 to update
413001026 - New town code entered, press F12 to update
413001015 - Enter required keys
413001017 - Duplicate REG
413001020 - Duplicate VIN

**Host program(s)**

**UHU4023P**

413001011 - Policy number already on file
413001006 - LR error
413001007 - DML error
413001016 - No more vehicles

**UHU4024P**

UHU4024L00001102 - Record obtained
UHU4024L00001202 - Other actions
UHU4024L03261101 - Record not found
UHU4024L00009010 - Modified record OK
UHU4024L00001001 - OFNS-NCSH record stored
UHU4024L03261301 - VEHR not found
UHU4024L00001401 - SLR-NCSH record stored
UHU4025P
413001008 - Resolve duplicates
413001005 - Record not found
413001010 - No registration
413001006 - LR error
413001002 - Update successful
413001016 - No more VEHC
413001017 - Duplicate registration
413001020 - Duplicate VIN

Map
UGU1020M

Copy books
UGZCOMMY (ALL)
UIU3010Y (UGU1021P)
UIU4120Y (ALL)
UGU4PICY (UGU1021P)
UIU3020Y (UGU1022P)
UHUUILOGY (UHU1023P)
UHRREGSY (UHU1023P)
UILXCNST (UHU4025P)
MREYIOWK (UHU4025P)
MUMYVEHC (UHU4025P)

External name
UPIC

Internal name
UU21, UU20

AVAILABLE FUNCTION KEYS:

F1  - End session
F2  - UMS menu
F3  - Not available
F4  - Not available
F5  - Not available
F6  - Not available
F7  - Scroll to the first page
F8  - Scroll to the second page
F9  - Not available
F10 - Not available
F11 - Not available
F12 - Update
UPMV Function

TITLE: MULTIPLE VEHICLE AMEND

DESCRIPTION: The UPMV screen is used to inquire on, add, or delete the vehicles associated with an insurance policy. The required key data fields are policy type, insurance company code, policy number, and policy effective date. The screen will display policy information, policyholder name, and all the vehicles, registrations, and owners currently insured on the policy. Optional key data fields are plate type and registration number. When these are keyed, the screen will display vehicles starting with the keyed registration.

To process a policy vehicle inquiry, an action code of “I” is entered with the required and optional key data fields.

To process a policy vehicle add, an action code of “A” is entered with the required key data fields. The optional key data fields are not allowed on an add. The 12 vehicle detail lines will be cleared and the new vehicles may be added. The keys that may be entered are VIN or plate type, registration, and plate color. The keys entered on each detail line must be unique and must exist on the database. If there are duplicate records on the database for the VIN or registration entered, a screen of duplicate key choices is displayed to allow the user to select the desired duplicate. The vehicle effective date is a required field, while the premium town code is optional. The vehicle effective date must be within the policy effective dates. When updates are applied to the database for an added vehicle, this date will become the vehicle’s beginning coverage date. If a premium town code is entered and it is different from the registration’s town code, an informational message is displayed. The F12 key is used to apply the updates to the database. Twelve vehicles may be added at one time. Once these vehicles have been added to the database, pressing F8 or Enter will clear the vehicle detail lines and allow 12 more vehicles to be added.

To process a policy vehicle delete, an action code of “D” is entered with the required and optional key data fields. The vehicle effective date initially displayed, (i.e. the vehicle’s beginning coverage date), is cleared and the user must enter the effective date of the delete. The entered vehicle effective date must be within the policy effective dates and not less than the vehicle’s beginning coverage date. When updates are applied to the database for a deleted vehicle, this date will become the vehicle’s ending
coverage date. The F12 key is used to apply the updates to the
database. Once the database updates are complete, the updated
vehicle information will automatically be displayed.

Security will only allow insurance companies to update their own
policies. However, inquiries may be performed on policies for any
company.

The F3 key provides access to a limited secondary session. The
Registration/Title Inquiry screen (URI) will be used in this
secondary session example. After the user has displayed the
information for a policy on the UPMV screen, which includes the
VINs and registrations connected with that policy, they change the
function code from UPMV to URI, place the cursor on the desired
VIN/Registration detail line, and press F3. This will transfer them
to the URI screen and display the information for the selected
registration. To return to the UPMV screen, the user may press the
Clear or F1 key. After being returned to the UPMV screen, a
message is displayed saying that the primary session has been
resumed.

The F6 key can be used for screen hopping. The user will be able
to screen hop to screens that have a policy, license, or FID number
as their primary key. For example, the user may F6 to UPTH
using the policy displayed on UPMV or F6 to UPH using the
policyholder’s license or FID number.

The F7 and F8 keys can be used for paging backward and forward
if the policy covers more than 12 vehicles.

The F9 key can be used as a quick way to jump to the UPA screen.
If a policy has been displayed and the user presses the F9 key
while the cursor is located anywhere on the first 12 lines of the
screen, the user will be screen hopped to UPA. It is important to
note that if the user is on a screen that allows limited secondary
sessions (F3 key) and they transfer to UPA and then use the “V”
action code to transfer to UPMV, they must use F9 to return to
UPA. If the Clear or F1 key is pressed, they will bypass the UPA
screen and be returned to the primary session. The F9 key may
also be used to resolve duplicates from Registration and VIN scroll
screens.

The F12 key will perform database updates.
COMPONENTS:  

*Guest input program*  
UGU1141P  
300310003 - Screen hop error  
411141001 - Invalid program entry  
411141002 - Enter vehc  
411141003 - Invalid VIN  
411141004 - VEHC key required  
411141005 - Invalid effective date  
411141006 - Nonunique reg  
411141007 - Nonunique VIN  
411141008 - F12-Message  
411141009 - No vehicles  
411141010 - Required field missing  
411141011 - Less bcov date  
411141012 - No deletes  
411141013 - F7 not allowed  
411141014 - First page  
411141015 - Last page  
411141016 - F12 after update  
411141017 - F9 surrogate error  
411141018 - Town F12 message  
411141019 - Invalid F4 surrogate  
411141020 - Enter keys  
411141021 - No reg add on  
411141022 - Invalid insurance number  
411141023 - Invalid character  
411141024 - Policy missing  
411141025 - Effective date missing  
411141026 - Maximum pages  
411141027 - Delete message  
411141028 - No amends  
411141029 - Empty page  
411141030 - Action missing  
411141031 - Policy type missing

*Guest output program*  
UGU1142P

*Host programs*  
UHU1143P, UHU1144P

*Map*  
UGU1140M

*LxTable*  
UGU1140T
Copy books
UGZCOMMY (UGU1141P, UGU1142P)
UHZCOMMY (UHU1143P)
UIU114AY (UGU1141P)
UIU114BY (UGU1141P, UGU1142P, UHU1143P)
UIU114CY (UHU1143P, UHU1144P)
UGZ2RFLY (UGU1142P)
UILXCNST (UGU1141P, UGU1142P, UHU1143P)
UHUILOGY (UHU1143P, UHU1144P)
MREYIOWK (UHU1143P)
MREYEDIT (UHU1143P)
MSUYTRAN (UHU1144P)

External name
UPMV

Internal name
PA09

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Limited secondary session
F4 - Not available
F5 - Not available
F6 - Screen Hopping
F7 - Scroll backward
F8 - Scroll forward
F9 - Screen hop to UPA
  Resolves duplicates from Registration and VIN scroll screens
F10 - Not available
F11 - Not available
F12 - Update
UPOI Function

TITLE: POLICY OPERATOR INQUIRY

DESCRIPTION: The UPOI screen is used to request operator information by operator license or FID number and policy effective date. The required fields are insurance company code, vehicle number, effective date, and operator license or FID number.

If the policy number is entered, then the coverage code and market indicator are required. If the license state is not entered, the default is ‘MA.’ If more than one operator is entered, then at least one of the operators must be marked as the policy holder.

The user has the option to start a second session by changing the function code and pressing F3. The information on the screen is carried over to the requested screen. If F3 is pressed again, the user is returned back to the UPOI screen. By using the secondary session, the user can be doing two functions at once.

When a tape of a new UMS release is sent to the insurance companies, the companies are given 60 days to make changes to the guest software and maps to make the applications more suitable to their individual business needs. For this reason, it is necessary to maintain two versions of the software. During those 60 days, the old, unchanged version of UPOI may be accessed by entering function code “TPOI,” while function code “UPOI” will execute the recently modified modules. After the 60-day period, TPOI will no longer be available.

Note: A typical use of this screen would be to inquire on the operators that are going to be added to a new policy.

COMPONENTS: Guest input program
UGU3011P
- 412001001 - BAD CALL TO DATE CONVERTER
- 412001002 - NO SURROGATE FOR F9
- 412001004 - PLEASE ENTER REQUIRED KEYS
- 412001008 - POLICY NUMBER CONTAINS INVALID CHARACTERS
- 412001009 - KEY PRESSED IS INVALID
- 412001010 - DUP LICENSE NUMBER KEYED
- 412001011 - EXPIRE DATE INVALID
- 412001012 - EFFECTIVE DATE INVALID
- 412001013 - REQUIRED FIELD MISSING, PLEASE
ENTER
412001014 - AT LEAST ONE OPERATOR IS REQUIRED
412001015 - POLICY HOLDER MUST BE INDICATED
412001027 - F9 DUP ONLY
412001028 - INSURANCE COMPANY NUMBER INVALID, PLEASE RE-ENTER
412001029 - VEHICLE NUMBER INVALID, PLEASE RE-ENTER
412001003 - ENTRY NOT VALID
412001039 - ONLY ONE PH
412001042 - COMM POLICY EFF DATE INVALID

Guest output program
UGU3012P
412001018 - INQUIRY PROCESS COMPLETE
412001016 - A RECORD MATCHING ENTERED KEY(S) WAS NOT FOUND
412001025 - DUPLICATES EXIST PRESS ENTER TO RESOLVE
412001026 - DUP LICENSE NUMBER KEYED, PLEASE CORRECT
412001030 - OUT OF STATE LIC # NOT FOUND PROCEED WITH UPDATE
412001031 - MASS/LIQUOR ID FOUND - PROCEED WITH UPDATE
412001033 - OUT OF STATE LIC # DUPLICATES FOUND
2001040 - DUPLICATE FID FOUND
2001041 - FID NOT FOUND

Host program(s)
UHU3013P
412001016 - A record matching entered keys was not found
412001017 - Database error
412001032 - Signon not authorized for ins co inquiry
412001041 - Record not found
412001042 - Database error
412001050 - Invalid character entered
412001051 - Exp date invalid
412001052 - Eff date invalid
412001053 - Eff date missing
412001054 - License FID required
412001055 - Phone number required

Appendix G – Functions Descriptions

412001056 - Insurance number invalid
412001057 - Only one phone
412001058 - Commercial policy effective date invalid
412001059 - Missing policy type value
412001060 - Invalid state for license
412001061 - Enter expiration date
412001062 - May not use reserved FID
412001063 - Bad effective date conversion (gregorian)
412001064 - Bad current date conversion
412001065 - Bad effective date conversion (julian)
412001066 - Bad comparison date conversion

UHU3014P
412001001 - Bad date conversion
412001005 - Database error
412001032 - Signon not authorized for ins co inquiry
412001037 - Years expired invalid
430140001 - Bad link service module 1
430140002 - Bad link service module 3
430140003 - Bad link logging module

Map
UGU2010M

Copy books
UGZCOMMY  (ALL)
UHUIMOY   (UHU3014P)
UHZSTATC  (UHU3014P)
MREYIOWK  (UGU3012P)
UIU3010Y  (ALL)
UGZ2RFLY  (UGU3012P)
MREYIOMW  (UGU3012P)
UGU3POIY  (UGU3011P)
MREYIORC  (UHU3014P)
UHU3DIPY  (UHU3014P)

External name
UPOI

Internal name
UU25
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Limited secondary session (requires function change)
F4 - Not available
F5 - Not available
F6 - Not available
F7 - Not available
F8 - Not available
F9 - Duplicate resolution from license or FID scroll screens
F10 - Not available
F11 - Not available
F12 - Not available
### UPTH Function

**TITLE:** POLICY TRANSACTION HISTORY INQUIRY

**DESCRIPTION:**

The UPTH function is used to display policy history information for a policy. To use this screen, the insurance company number, policy number, and policy effective date must be entered. Also, a “C” (Commercial) or “P” (Private) needs to be added to the POL TYPE field.

The detail lines on this screen represent each policy history entry recorded for a policy and will be in ascending transaction date order. Multiple pages may be displayed by using the standard F7 and F8 for page backward and forward.

This screen can “hop” to UPA, UPMV, or UPH by changing the function code to the desired screen and pressing the F6 key.

When a tape of a new UMS release is sent to insurance companies, they have 60 days to make changes to the guest software and maps to make the applications more suitable to their individual business needs. For this reason, it is necessary to maintain two versions of the software. During these 60 days, the old version of UPTH may be accessed by entering the function code “TPTH,” while the function code “UPTH” will execute the recently modified modules. After the 60-day period, TPTH will no longer be available.

**NOTE:** A typical use of the UPTH screen would be to find detailed policy history information related to a particular policy.

**COMPONENTS:**

*Guest input program*

UGU2131P

415001001 - Entry not valid
415001002 - Enter required fields
415001003 - Policy number contains invalid characters
415001004 - Policy number missing or invalid
415001005 - Insurance company number invalid
415001006 - Policy effective date missing
415001007 - F7 is not valid
415001008 - No surrogate for F4
415001009 - End of set - no more records
415001010 - Policy type missing

Guest output program
UGU2132P
  415001009 - No more records
  415001015 - F8 For more records

Host program(s)
UHU2133P
  421330001 - Bad link
  415001020 - LR error - work-plcy-request invalid
  415001021 - A record matching entered keys was not found
  415001022 - LR error
  415001023 - No history records found for policy
  415001024 - LR error using dbkey

Lr
UHU1133L

Map
UGU1130M

Lx table
UGU1130T

Copy books
UGZCOMMY (UGU2131P, UGU2132P)
UHZCOMMY (UHU2133P)
UGU213AY (UGU2131P, UGU2132P)
UGU213BY (UGU2131P)
UIU213CY (UGU2132P, UHU2133P)
UHUILOGY (UHU2133P)
MREYIORC (UHU2133P)

External name
UPTH

Internal name
UU26
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Not available
F4 - Not available
F5 - Not available
F6 - Screen hopping
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available

NOTE: F6 requires function change
**URN Function**

**TITLE:** NON-INDIVIDUAL (COMPANY/CORPORATE) NAME INQUIRY

**DESCRIPTION:** The URN function is used to request license or registration information by company/corporation name. The key fields are company name, active plates indicator, exact company name match indicator, and company only indicator. Only the company name is required, and this entry may be as short as one character (for example, locate all companies whose names begin with a particular letter). The active plates indicator, exact company name match indicator, and company only indicator are Y (yes) or N (no) fields, with a system default of N. In order to activate these fields, the operator must type over the N on the screen with a Y.

If the active plates indicator is entered as a Y, only active plate numbers are displayed for the requested company name; otherwise, all plate numbers for the requested company are selected (active, expired, revoked).

If the exact company name match indicator is entered as a Y, all characters entered in the company name field are used as a search argument; otherwise, only the first twenty characters are used.

If the company only indicator is entered as a Y, only the company name and address displays on the screen; registration information is suppressed.

**COMPONENTS:**

*Guest input program*
UGR0010P
203001001 - Please enter CORP/CO name
203001002 - No records for this key on database
203001003 - Enter Y or N for active plate
203001004 - Enter Y or N on exact CORP/CO match
203001005 - Enter Y or N for CORP/CO only
203001007 - F8 Invalid - inquiry must be done first
203001008 - F7 Invalid - no prev screen encountered
203001010 - F8 Invalid - improper cursor positioning
203001011 - F8 Invalid - end of set encountered

*Guest output program*
UGR0011P
203001006 - End of set encountered
203001012 - Enter F8 to begin scroll
203001013 - Enter F8 to continue to scroll forward
203001014 - Maximum number of reads -
press F8 to continue

Host program(s)
UHR0010P

203001002 - No records for this key on database

Map
UGR0010M

Copy books
UGZCOMMC (UGR0010P, UGR0011P)
DFHATTRB (UGR0010P)
DFHAID (UGR0010P)
UHZCOMMC (UHR0010P)
IDMS LR UHR0010L-LR (UHR0010P)
IDMS SUBSCHEMA BINDS (UHR0010P)

External name
URN

Internal name
RN01

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Rescroll
F12 - Not available

NOTE: F4 requires function change
URSN Function

TITLE: Registration/VIN Scroll/Name

DESCRIPTION: The URSN function is used to request and display a list of vehicle registrations or vehicle VINs by name. The key fields (fields to be entered) are last name, first name, middle name, date of birth, or registration status. The last name is the only required field, and this entry may be as short as one character. The first name, middle name, date of birth, and registration status are optional but may be used to narrow the search. In addition to the key fields, the VIN display field determines whether vehicle registration or VIN information will be displayed.

Note: A typical use of this screen would be to find the registration of an individual when only a name or part of a name is available.

COMPONENTS: Guest input program
UGR1091P

Guest output program
UGR1092P

Host program
UHR1093P

Map
UGR1090M

Copy books
UGZCOMMY (ALL)
UGR1090Y (UGR1091P, UGR1092P)
UHUILOGY (UHR1093P)
UIR1090Y (UGR1092P, UHR1093P)
UIR1RSNY (UGR1091P)

External name
URSN, TRSN

Internal name
UR07
AVAILABLE FUNCTION KEYS:

- F1 - End session
- F2 - UMS Menu
- F3 - Not available
- F4 - Select from scroll screen (requires function change)
- F5 - Not available
- F6 - Not available
- F7 - Page backward
- F8 - Page forward
- F9 - Select from scroll screen (requires function change)
- F10 - Not available
- F11 - Rescroll
- F12 - Not available
**URSR Function**

**TITLE:** REGISTRATION SCROLL/REGISTRATION

**DESCRIPTION:** The URSR function is used to request and display a list of vehicle registrations by registration number. The key fields (fields to be entered) are plate type, registration number, plate color, and registration status. The registration number is the only required field. This entry may be as short as 1 character (for example, to display all registrations whose registration number begin with the number 4). The plate type, plate color, and registration status are optional but may be used to narrow the search.

NOTE: A typical use of this screen would be to find the registration of an individual if all or part of the registration number is known.

**COMPONENTS:**

- Guest input program
  UGR11241P

- Guest output program
  UGR1242P

- Host program
  UHR1243P

- Map
  UGR1240M

- Copy books
  UGZCOMMY (ALL)
  UGRIRSRYY (UGR1241P)
  UHUILOGY (UHR1243P)
  UIR1240Y (UHR1243P, UGR1241P, UGR1242P)
  UHRREGSY (UHR1243P)

- External name
  URSR

- Internal name
  UR02
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page forward
F8 - Page backward
F9 - Select from scroll screen
F10 - Not available
F11 - Rescroll
F12 - Not available

NOTE: F4 and F9 require a function change
URSV Function

TITLE: REGISTRATION SCROLL/VEHICLE IDENTIFICATION NUMBER (VIN)

DESCRIPTION: The URSV function is used to request and display a list of vehicle registrations by VIN. The key fields (fields to be entered) are VIN and registration status. The VIN is the only required field, and this entry may be as short as one character. The registration status is optional but may be used to narrow the search.

NOTE: A typical use of this screen would be to find the registration of an individual if all or part of the VIN is known.

COMPONENTS: 

Guest input program
UGRI101P
310001001 - End of set encountered
310001002 - Beginning of set encountered
310001003 - Status keyed is invalid
310001004 - Entry not valid
310001005 - Please enter keys
310001006 - VIN missing - please enter
310001007 - Enter key valid with key changes only

Guest output program
UGRO101P
310001008 - F8 for more records
310001009 - End of set encountered, no more records
310001015 - Duplicate encountered

Host program
UHRI101P
310001010 - Logical record error
310001011 - IDMS error
310001012 - A record matching entered key was not found
310001017 - Response not normal

Map
UGR101UM

Copy books
UGZCOMMY (ALL)
UHRO101Y (UHRI101P)
UGUIPOIY (UHRI101P)
UIRO101Y (UGRO101P)
UIRI101Y (UGRI101P, UGRO101P)
UHIUILOGY (UHRI101P)

External name
URSV

Internal name
UR03

AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Not available
F4 - Select from scroll screen
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Select from scroll screen
F10 - Not available
F11 - Rescroll
F12 - Not available

NOTE: F4 and F9 require a function change
UVH Function

TITLE: VEHICLE POLICY HISTORY INQUIRY

DESCRIPTION: The UVH function is used to display policy information related to a registration or vehicle identification number (VIN). The user must enter either the registration information (plate type, registration number, and color) or the VIN to use this screen. If a duplicate registration key is entered, the user will be transferred to URSR and prompted to cursor select the correct registration number. If a duplicate VIN is entered, URSV will execute, and the user will be prompted to cursor select the correct VIN.

The detail lines on this screen represent each policy that the keyed vehicle/registration was covered by and when it was covered. The detail lines are sorted in descending order by policy begin coverage date. Multiple pages may be displayed on this screen by using the standard F7 and F8 for page forward and back.

This screen can be “hopped” to/from URI using F6 or from URSR/URSV/URSN with an F4 cursor select. In addition, the F3 key provides access to a limited secondary session. Secondary session is invoked by taking the following steps. After displaying the policies for a registration or VIN using the UVH screen, change the function code from UVH to UPA, cursor select the desired policy detail line, and press F3. This will transfer the user to the UPA screen and display the information for the desired policy. At this time, the user is in inquiry mode, but may change the action code and press the Enter key to change the mode to update. To return to the UVH screen, press the Clear key or the F1 key and resume the primary session.

The F11 key is used for rescrolling. When the user has been transferred to the UVH screen by pressing the F4 or F9 key on a scroll screen, the user may return to the scroll screen in order to make a new selection. This is done by changing the function code to the appropriate scroll screen’s function and pressing the F11 key.

When a tape of a new UMS release is sent to insurance companies, they have 60 days to make changes to the guest software and maps to make the applications more suitable to their individual business needs. For this reason, it is necessary to maintain two versions of the software. During these 60 days, the old version of UVH may be accessed by entering the function code “TVH,” while the
function code “UVH” will execute the recently modified modules. After the 60-day period, TVH will no longer be available.

NOTE: A typical use of the UVH screen would be to find policy information related to a registration or VIN.

COMPONENTS:

Guest input program
UGU3111P
414001001 - Plate type or VIN equal spaces
414001002 - No surrogate found - F4
414001003 - F7 not valid
414001004 - No more records
414001005 - Entry not valid
414001025 - Reference not number

Guest output program
UGU3112P
414001009 - F8 for more
414001004 - No more records
414001013 - Policy not found

Host program
UHU3113P
414001011 - Registration not found
414001012 - VIN not found
414001014 - LR error
414001015 - More than 120 policies
414001016 - Paging error
414001020 - Lookaside invalid function
414001021 - Lookaside surrogate invalid
414001022 - Lookaside length error
414001027 - LR error
414001028 - LR error
414001029 - LR error
414001030 - LR error
414001031 - LR error
414001032 - No section 1032
414001033 - No registration
414001034 - 1 policy not found
431130001 - Bad link on lookaside
431130002 - Bad link logging mod

LR
UHU2113L
Map
UGU3110M

Lx table
UGU3110T

Copy books
UGZCOMMY (UGU1111P, UGU1112P)
UHZCOMMY (UHU1113P)
UGU111AY (UGU1111P, UGU1112P)
UGU111BY (UGU1111P)
UIU111CY (UGU1112P, UHU1113P)
UGZ1RFLY (UGU1112P)
UHUILOGY (UHU1113P)

External name
UVH

Internal names
VH07, VH08

AVAILABLE FUNCTION KEYS:

F1  - End session
F2  - UMS Menu
F3  - Limited secondary session
F4  - Select from scroll screen
F5  - Not available
F6  - Screen hopping
F7  - Page backward
F8  - Page forward
F9  - Resolve duplicates from Registration or Vehicle scroll screens
F10 - Not available
F11 - Rescroll
F12 - Not available

NOTE: F3, F4, and F6 require function change
VT Function

TITLE: VEHICLE TITLE HISTORY

DESCRIPTION: The VT function is used to request and display current and historical title and registration information by the Vehicle Identification Number (VIN). The key field (field to be entered) is the VIN, which is the only required field.

NOTE: A typical use of this screen would be to find the current and historical titles and registrations attached to a vehicle.

Multiple pages may be displayed on this screen by using the standard F7 and F8 for page forward and back.

This screen can be transferred from a UMS screen (eg. RH, T1A) or an ALARS screen (eg. TH, RVV) if a vehicle key is passed. The screen can transfer to another screen using either the vehicle, title, or registration information. Pressing F6 transfers to another screen with the vehicle information. Positioning the cursor to a detail line and pressing F4 transfers to another screen with either the title or registration.

This screen resolves duplicate VIN numbers by transferring to the URSV screen which will display the duplicate VINs, owners, registrations, and the current status. By positioning the cursor on a detail line and pressing F9, URSV will transfer back to the VT screen and display the current and historical vehicle information for the entry selected.

COMPONENTS:

Guest input program
UGR3261P

327001001 - Entry not valid
327001002 - Enter keys
327001003 - No surrogate F4
327001004 - Beginning of set encountered
327001005 - End of set
327001006 - Leading spaces

Guest output program
UGR3262P

327002001 - F8 for more
327002002 - No more records
Host program
UHR3263P
  327003001 - VIN not found
  327003002 - No records for key
  327003003 - No more data
  327003004 - LR error
  327003005 - Bind error
  327003006 - Finish error
  327003007 - Vehicle information
  327003008 - DML error

Map
UGR3260M

Copy books
UGZCOMMY (UGR3261P, UGR3262P)
UHZCOMMY (UHR3263P)
UGR326AY (UGR3261P, UGR3262P)
UGR326BY (UGR3261P)
UIR326CY (UGR3262P, UHR3263P)
UHUILOGY (UHR3263P)
MREYBNME (UHR3263P)
MREYBADR (UHR3263P)
MREYBADW (UHR3263P)
MREYBNMW (UHR3263P)
MREYBADP (UHR3268P)
MREYBNMP (UHR3268P)

External name
VT

Internal name
VT03
AVAILABLE FUNCTION KEYS:

F1 - End session
F2 - UMS Menu
F3 - Not available
F4 - Screen transfer using cursor select
F5 - Not available
F6 - Screen transfer using keyed vehicle
F7 - Page forward
F8 - Page backward
F9 - Not available
F10 - Not available
F11 - Rescroll
F12 - Not available
ULP Function

TITLE: UMS PERSON NAME SCROLL

DESCRIPTION: The ULP function is used to request license or registration information by name. The key fields (fields to be entered) are last name, first name, middle name, and date of birth. Only the last name field is required, and this entry may be as short as one character (for example, locate all license holders whose last names begin with a particular letter). The first name, middle name, and date of birth entries are optional and used to narrow the search (for example, when locating John Q. Public born 10/30/41, the program returns only one entry).

COMPONENTS:

Guest input program
UGL0030P
201001001 - Invalid last name, please reenter
201001002 - Invalid first name, please reenter
201001003 - Invalid middle name, please reenter
201001004 - Invalid date of birth, please reenter
201001005 - Beginning of set encountered
201001006 - End of set encountered
201001007 - Enter name and date of birth, press enter to begin
201001008 - F7 Invalid - Inquiry must be done first
201001009 - F8 Invalid - Inquiry must be done first
201001010 - Last name is required

Guest output program
UGL0031P
201001006 - End of set encountered
201001011 - More records on database - use F8
201001012 - More records on database - use F8
201001013 - Maximum number of reads - press F8 to continue

Host program(s)
UHL0040P

Map
UGL0010M

Copy books
UGZCOMMC (UGL0030P, UGL0031P)
UHZCOMMC (UHL0040P)
IDMS LR UHL0040L-LR (UHL0040P)
DFHAID (UGL0030P)
IDMS SUBSCHEMA BINDS (UHL0040P)

*External name*
ULP

*Internal name*
LI03

**AVAILABLE FUNCTION KEYS:**

F1 - Main Menu
F2 - UMS Submenu screen
F3 - Not available
F4 - Select from scroll
F5 - Not available
F6 - Not available
F7 - Page backward
F8 - Page forward
F9 - Not available
F10 - Not available
F11 - Not available
F12 - Not available

**NOTE:** F4 requires function change
RA Function

TITLE: REGISTRATION AMEND

DESCRIPTION: This screen is only available to non-remote sites or at town offices. The RA Function is used to process changes or renewals for Registrations. Entering a valid registration number or VIN retrieves information on the screen. If a duplicate VIN or registration is entered, a second screen will appear (USRS or USRV) which will allow the user to select the correct record by placing the cursor on the record and then hitting the F9 key to return to the RA screen.

COMPONENTS: 

Guest Input Program
UGR1271P
- 328001002 - NRES LINK ERROR
- 328001003 - NO CASH POSTING
- 328001015 - BAD NRES RC MSG

Guest Output Program
UGR1272P

Host Programs:
UHR1273P
- 328003000 - Address Problem
- 328003001 - Bad Date Load
- 328003002 - No Renew Msg
- 328003003 - No VEHC Action
- 328003004 - Action required
- 328003005 - Missing action msg
- 328003006 - F12 to update
- 328003007 - Print success
- 328003008 - Init please continue
- 328003009 - Invalid NOWN PERS
- 328003010 - Invalid NUM occur
- 328003011 - Inv IA action
- 328003014 - Batch success
- 328003015 - Duplicate batch number
- 328003016 - Closed batch print
- 328003017 - Bad print value
- 328003018 - Bad SYSID load
- 328003019 - Owner blocked
- 328003020 - INCP not found
- 328003021 - Drawer problem cash pgm
- 328003022 - Invalid RC cash pgm
328003023 - Bad cash table search
328003024 - Msg VEHR not found

**MAP:**
UGR1270M

**Table:**
UGR1270T

**Copy Books:**
UGZCOMMY - Host Commarea (UHR1273P, UGR1272P, UGR1271P)
UIR127AY - GSA/ Host-to-Guest Block (UHR1273P, UGR1272P, UGR1271P)
UHR1273Y - RA Host Commarea (UHR1273P)
UILXCNST - LX Table Constants (UHR1273P, UGR1271P, UGR1272P)
MREYCSHC - Cash Code Table (UHR1273P)
UHRCKNWY - Owner Blocking Action (UHR1273P)
MREYURP - URI Print Layout (UHR1273P)
MREYBADR - Address Work Area (UHR1273P)
MREYBNME - Name Compression Area (UHR1273P)
MREYPPNM - Plate Names Table (UHR1273P)
MCCYCCRD - Credit Card Payment Interface (UHR1273P)
UHUILOGY - Host Message Logging Parm (UHR1273P)
WAAYCDAT - Waasdate Area (UHR1273P)
UIGSPPRT - Online Reg/ Title Print Comm Area 1 (UHR1273P)
UIGSPUR1 - Online Reg/ Title Print Comm Area 2 (UHR1273P)
UIR1FLGY - Data Management Control Flags (UHR1273P)
MREYEDIT - Plate Type/ Reg Format Editor (UHR1273P)
UHR1EXLY - Excise Tax Record Builder (UHR1273P)
UIR1COMR - ALARS Commarea (UGR1271P)

**External Name:**
RA

**Internal Name:**
RA01
RA02
RA03
AVAILABLE FUNCTION KEYS:

F1 - End Session
F2 - UMS Menu
F3 - Not Available
F4 - Not Available
F5 - Not Available
F6 - Not Available
F7 - Not Available
F8 - Not Available
F9 - Not Available
F10 - Not Available
F11 - Not Available
F12 - Update
Appendix H: CICS IDMS Network Relationships

NETWORK CONNECT OVERVIEW

------------

<table>
<thead>
<tr>
<th>AMDAHL 5890</th>
<th>AMDAHL 4745</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS/XA</td>
<td>SNI/</td>
</tr>
<tr>
<td>VTAM</td>
<td>GATEWAY</td>
</tr>
<tr>
<td>CICS</td>
<td>NCPS</td>
</tr>
<tr>
<td>IDMS</td>
<td>NETWORKS</td>
</tr>
<tr>
<td>TSO</td>
<td>]</td>
</tr>
</tbody>
</table>

Appendix H – CICS IDMS Network Relationships
CV10

-------------

*** MVT1CONN

BAOCICS1

-------------

CICS/OS2

BAOCICU1

-------------
---

**IB2WIBM1**

[---] IBM(IVANS)

[---] BAOCICG1

---

OLSJCIUN

[---] USING [---]===] BAOCICR1 [===]===

---

BAOCICU1

[---]===] LU 6.2

---

NDRT

[---] (ISC)

---

NO CV

[---]

---

MVS1CONN

*** MVU1CONN

---

OLSJCIMN

---

BAOCICM1

---

MODL

---

CV12

---

*** = CEDA GROUP NAMES

---

TRAINING ENVIRONMENT

---

OLSJCIWN

---

BAOCICM1

---

MODL

---

CV12

---

*** = CEDA GROUP NAMES
MRO ENVIRONMENT

------------

[ ]===] BAOCICM1 ] *** MVM1MRO
[ ] [ ]___________] SYSID=MRM1

[ ] OLSJCIUN ] [ ] USING ] [ ] MVM1MRO
[ BAOCICU1 ] [ ] LU 6.1 ] [ ] MVM1MRO
[ ] [ ] (MRO) ] [ ] MVM1MRO
[ ] [ ] [ ] [ ] MVM1MRO

*** MVU1MRO
SYSID=MRU1

------------

[ ]===] BAOCICT1 ] *** MVT1MRO
[ ] [ ]___________] SYSID=MRT1

[ ] OLSJCIEN ] [ ] USING ] [ ] MVT1MRO
[ BAOCICE1 ] [ ] LU 6.1 ] [ ] MVT1MRO
[ ] [ ] (MRO) ] [ ] MVT1MRO
[ ] [ ] [ ] [ ] MVT1MRO

*** MVR1MRO
SYSID=MRR1

------------

[ ]===] BAOCICP1 ] *** MVP1MRO
[ ] [ ]___________] SYSID=MRP1

[ ] OLSJCIE1 ] [ ] USING ] [ ] MVP1MRO
[ ] [ ] LU 6.1 ] [ ] MVP1MRO
[ ] [ ] (MRO) ] [ ] MVP1MRO
[ ] [ ] [ ] [ ] MVP1MRO

*** MVE1MRO
SYSID=MRE1

*** = CEDA GROUP NAMES
THE TMON V8.0 ENVIRONMENT

---
[ ] TMON8CSM ]
[ ] [ ]
[ ] INIT TST/ ]-----------------------[ ] OLSJCIUN ]
[ ] NEW RELS. ] [ ]
[ ]

---
[ ] TMONTCSM ]
[ ] [ ]
[ ] TEST CLUSTER]-----------------------
[ ] [ ]
[ ] [ ]
[ ]

---
[ ] TMONPCSM ]
[ ] [ ]
[ ] PROD CLUSTER]-----------------------
[ ] [ ]
[ ] [ ]
[ ]

* = TRIGGERS NIGHTLY DLS SWITCH PRIOR TO TMON MAINTENANCE
CICS - IDMS REGION RELATIONS

PRODUCTION ENVIRONMENT

-------------    -------------
| OLSJCIAN     | OLSJCIPN     |
|------------>  | ------------>
| BAOCICA1    | BAOCICP1    |
|            |            |
ISC         CV11         CV11         |
|            |            |
-------------    -------------
|              |              |
|              |              |
-------------    -------------
| M]            | M]        VIA    |               |
| VARIOUS       | R]        R]    |               |
| INS. COS.     | O]        O]    |               |
| (UMS)        |           |               |
-------------    -------------
|              | OLSJCIEN   | IBMIN        |
|              |            |              |
-------------    -------------
| M]            |            |               |
| VARIOUS       |            |               |
| INS. COS.     |            |               |
| (UMS)        |            |               |
-------------    -------------
| OLSJCIEN     |            |               |
| BAOCICA1    | BACKUP    TO RECOVER CV11, ALLOWS LAW |
| CV14        | ENFORCEMENT INQUIRY ACCESS. |
-------------
TEST AND TRAINING ENVIRONMENT

\[
\begin{array}{c|c}
\text{OLSJCIUN} & \text{OLSJCIMN} \\
\text{BAOCICU1} & \text{BAOCICM1} \\
\text{NDR} & \text{MRO} \\
\text{NO CV} & \text{CV12} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{OLSJCITN} & \text{OLSJCIRN} \\
\text{BAOCICT1} & \text{BAOCICR1} \\
\text{ISC} & \text{ISC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{IBMIN} & \text{IB1FNCIC} \\
\text{CV10} & \text{CV14} \\
\text{ISC} & \text{ISC} \\
\text{OLSJCISN} & \text{OLSJCIGN} \\
\text{BAOCICS1} & \text{BAOCICG1} \\
\text{CV13} & \text{UMS TEST} \\
\end{array}
\]