N-ARY LEVEL
IN THE SOFTWARE TEST VEHICLE
FOR THE INFOPLEX DATABASE COMPUTER

by

DAVID LUI

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE IN PARTIAL
FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF

BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITIUTE OF TECHNOLOGY

MAY, 1982

Signature
of
Author_____
Department of Electrical Engineering and Computer Science
May, 1982

Certified b_____
Professor Stuart Madnick
Thesis Supervisor

Accepted by_____
Professor David Adler
Chairman, Department Committee

CONTENTS

Chapter One

General Overview

1.1 Introduction

Infoplex is a database computer architecture having as its objective the support of large-scale information management with high reliability.(Ref.1) It aims to provide a solution to the problem of increasing loads, in terms of both throughput and volume of stored data, faced by today's and tomorrow's information processing needs.

Infoplex consists of a storage hierarchy, which supports a very large data storage system, and a functional hierarchy, which is responsible for providing all database management functions other than device management.

The Infoplex functional hierarchy is designed around a concept of hierarchical decomposition.(Ref.2) It is a discipline that helps in identifying the key functional modules that have minimal interdependencies and can be combined hierarchically to form a software system, such as an operation system or a database management system.

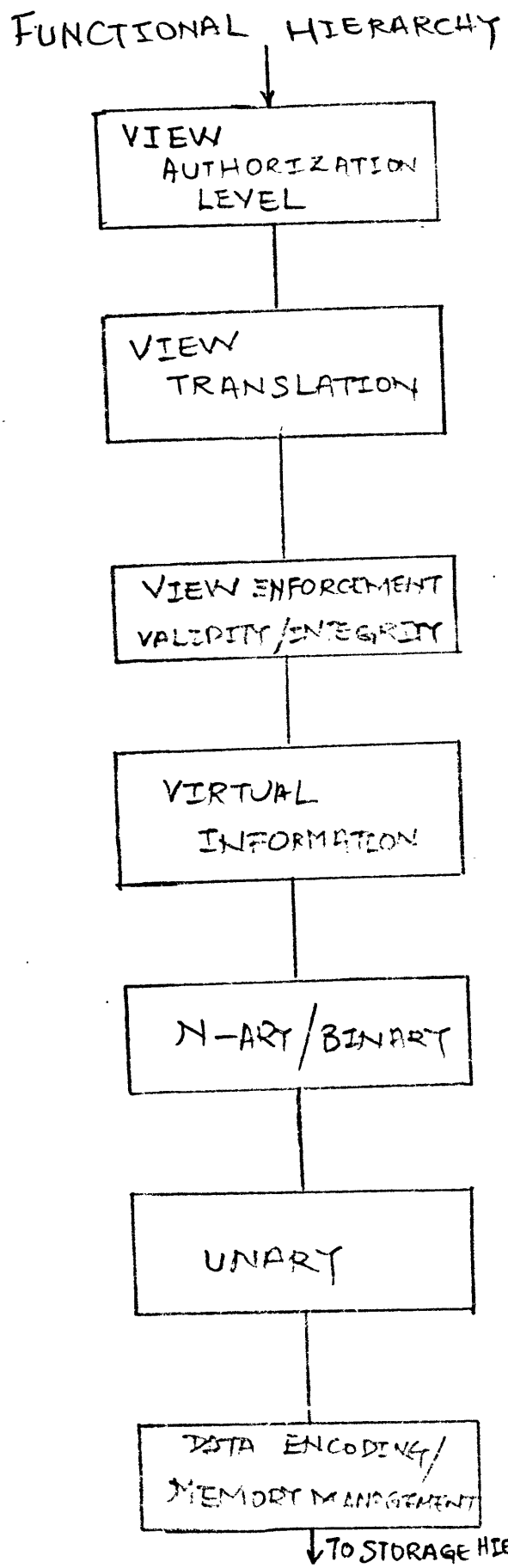

FUNCTIONAL HIERARCHY
VIEW AUTHORIZATION LEVEL
VIEW TRANSLATION
VIEW ENFORCEMENT VALIDITY/INTEGRITY
VIRTUAL INFORMATION
N-ARY/BINARY
UNARY
DATA ENCODING/ MEMORY MANAGEMENT
TO STORAGE HIERARCHY

Fig. 1

The idea of data abstraction is widely used in the functional hierarchy. The functional hierarchy takes care of accepting user's commands and doing the proper updating. According to the preliminary design, it is divided into ten levels each having its own data structures and its own functions. Communications between different levels take place in the form of control blocks. These blocks queue up and wait to be executed. The ten levels are i) View Authority Level ii) View Translation Level iii) View Enforcement Level iv) Validity/Integrity Level v) Virtual Information Level vi) N-ary Level vii) Binary Level viii) Unary Set Level ix) Data Encoding Level x) Memory Management Level. Refer to Fig.1.

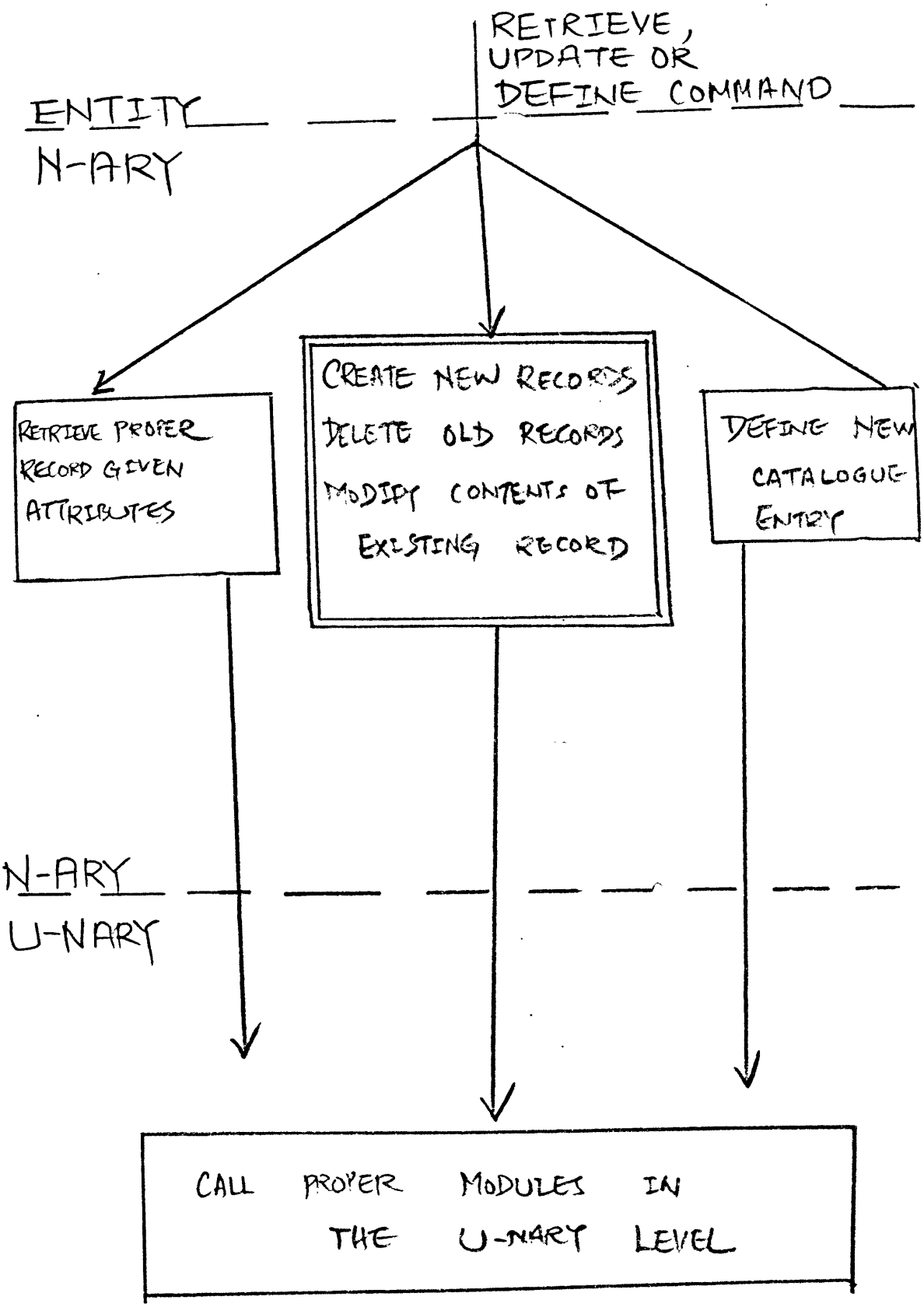
Before a hardware prototype is commissioned, a software simulation is to be implemented as a vehicle for validating the communication and functional algorithms in the preliminary design. Previous efforts in this software test vehicle(STV) project include an early version(Ref.3) and a later version(Ref.4) of the functional hierarchy, and the implementation of a control structure which emulates the multi-level multi-processing environment.(Ref.5) The work described in this proposal is part of an integrated effort to improve on the more recent version of the functional hierarchy. In particular my work involves implementation of the N-ary level.

1.2 Proposed Work

All the data in the N-ary level are grouped into Primitive sets(Psets) and are made to relate to each other through Binary Association sets(Bsets). The level above the N-ary level will map logical constructs such as entity sets and attributes onto Psets and Bsets. The N-ary level will, in turn, map Psets and Bsets onto record sets, or U-nary sets. It will, according to some algorithm, group related Psets into record sets, and implement additional associations through linkage field(s) in the records. The U-nary sets are implemented at the Unary Level, which directly supports the N-ary level.

The primary tasks of the N-ary Level include maintaining a catalogue of all the Psets and Bsets defined, updating of the data, and retrieval of data. Updating data expands into creating new records, deleting existing records and modifying the contents of specified records. Fig.2 shows the outline of the three subsystems of the N-ary Level and the interactions among the three.

My work is to implement the creation of new records, deletion of existing records and modifying the contents of specified records.


RETRIEVE, UPDATE OR DEFINE COMMAND
ENTITY N-ARY
RETRIEVE PROPER RECORD GIVEN ATTRIBUTES
CREATE NEW RECORDS
DELETE OLD RECORDS
MODIFY CONTENTS OF EXISTING RECORD
DEFINE NEW CATALOGUE ENTRY
N-ARY U-NARY
CALL PROPER MODULES IN THE U-NARY LEVEL

Chapter Two

Data Organization in the Entity and the U-nary Levels

2.1 Entity Level Concepts

At the Entity Level, data are organized by the binary network model. A visual presentation of our binary network model is shown in the following figure. There are four basic constructs. Primitive Elements represent some objects or facts in the real world.(Fig.2.1a) A primitive Set is a group of primitive elements that have similar generic properties and therefore are given a common group name, called a Primitive Set Name, or Pset_name.(Fig.2.1b) Binary associations are representations of some real world relationship among primitive elements from different primitive sets.(Fig.2.1c) A binary set is a group of binary associations that have similar generic properties(i.e., the incident primitive elements belong to the same primitive sets, and the associations have the same meaning). It is designated a pair of primitive set names and a pair of association names.(Fig.2.1d)

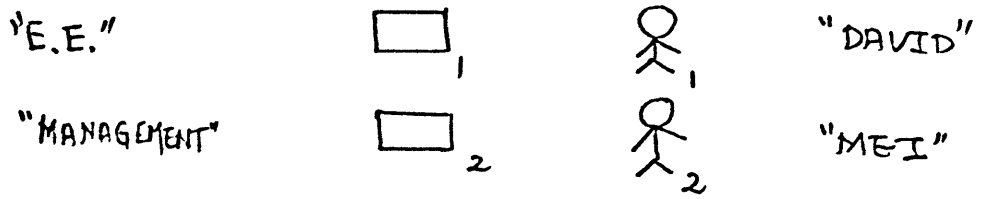


FIG. 2.1a : PRIMITIVE ELEMENTS

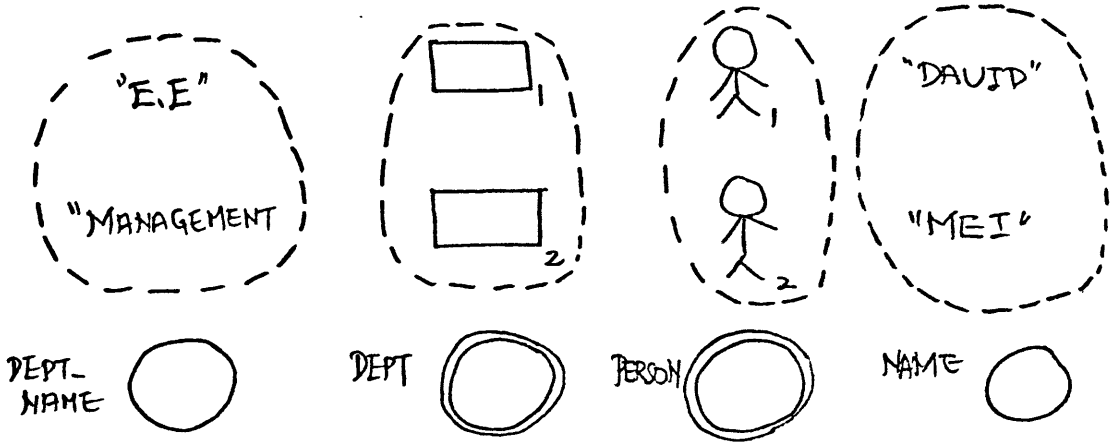


FIG. 2.1b: PRIMITIVE SETS (PSETS)

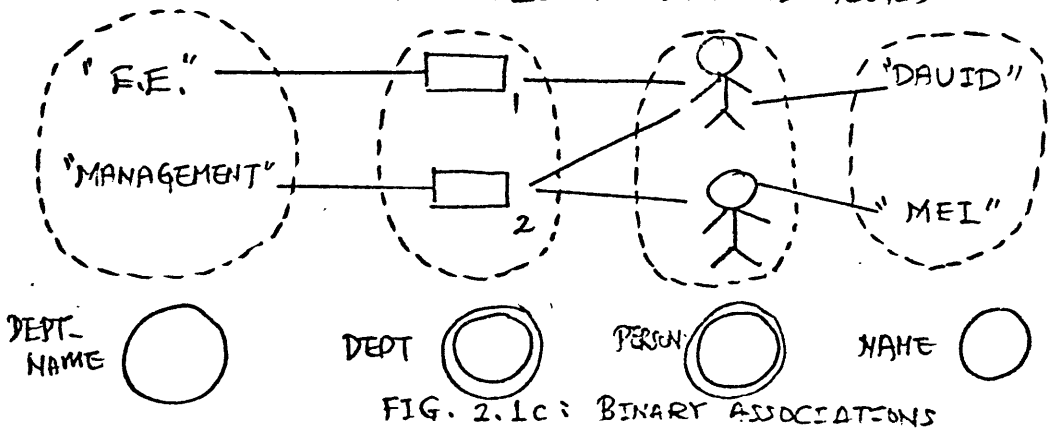


FIG. 2.1c : BINARY ASSOCIATIONS

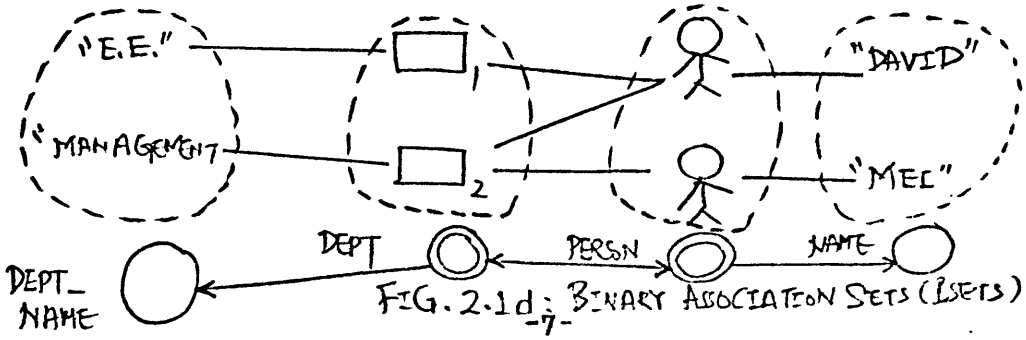
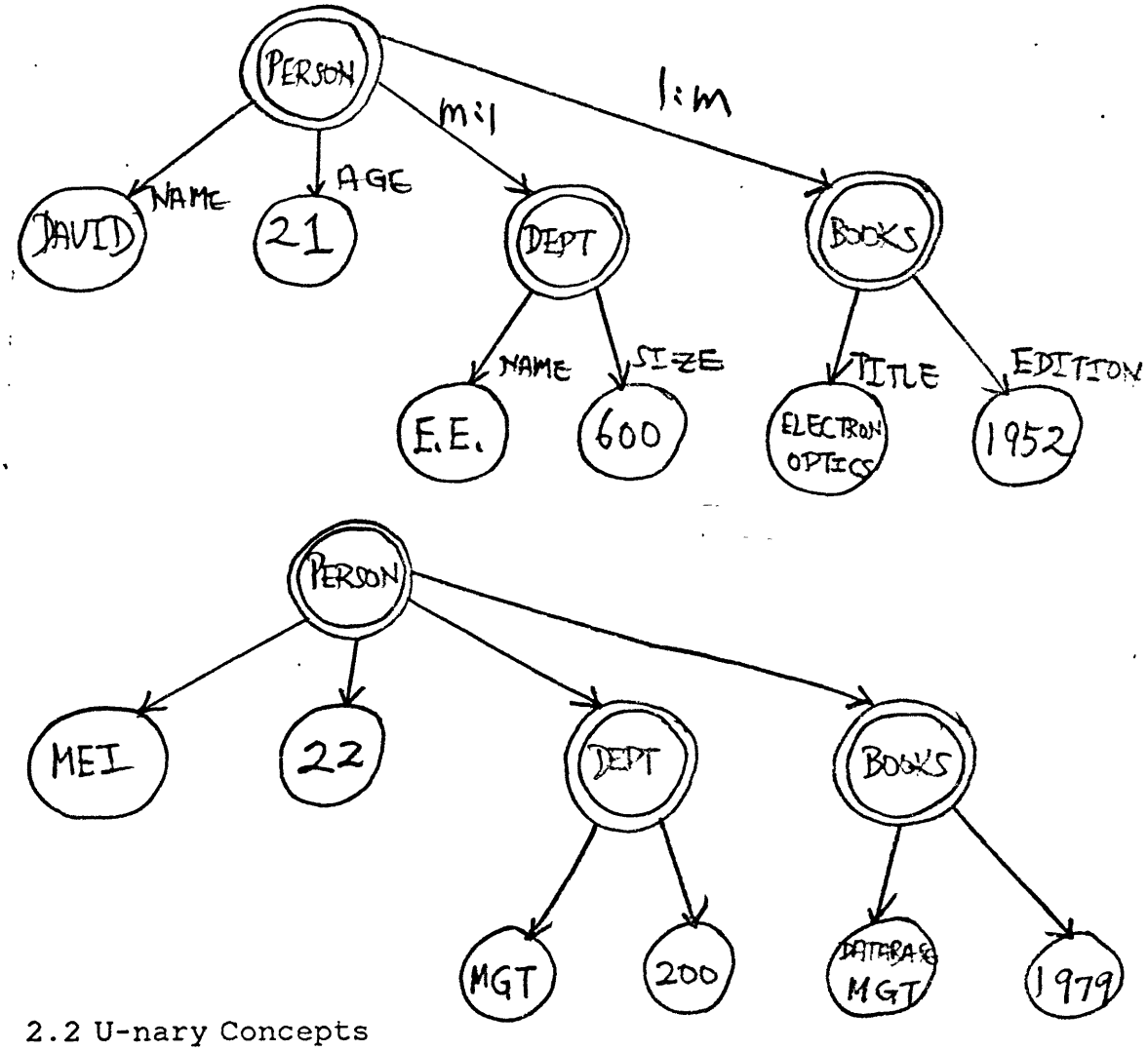


FIG. 2.1d : BINARY ASSOCIATION SETS (ISETS)

In Fig.2.1d, the upper portion (primitive elements and binary associations) represents the instance of the database, while the lower portion represents the schema of the database. Therefore, the schema of our Binary Network model is composed of primitive sets (also known as the 'nodes') and binary association sets (also known as the 'arcs').

Further classification of nodes and arcs: In a binary network schema graph, a node can be either an entity node or a value node. An entity node serves to tie all equally related value nodes or entity nodes together. By 'equally related' we mean that those nodes tied to this entity node are all direct attributes of the entity node, instead of 'derived' attributes. An entity node corresponds to any set of real world objects (tangible or intangible) that have some common set of attributes which are revealed by the node's binary connections to other value nodes or entity nodes. It's own identity is reflected by these associations; i.e., an instance of an entity node does not have any value or identity, and its designation is made through instances of its associated nodes. In a sense, the purpose of an entity node is to collect equally related binary associations to form a semantically clean N-ary association. Therefore an entity node is also referred to as an N-ary entity node. Those nodes that are not entity nodes are value nodes. Value nodes, in contrast to entity nodes, have values assigned to their instances.

Arcs can be specified by several parameters. It can be many-to-1(mto1), one-to-one(1to1), or one-to-many(1tom). A second parameter is 'E' or 'N' which means whether or not the child's recordid is embedded in the parent's record respectively. Fig.2.2 shows a couple of examples on how information is moulded into a data tree.



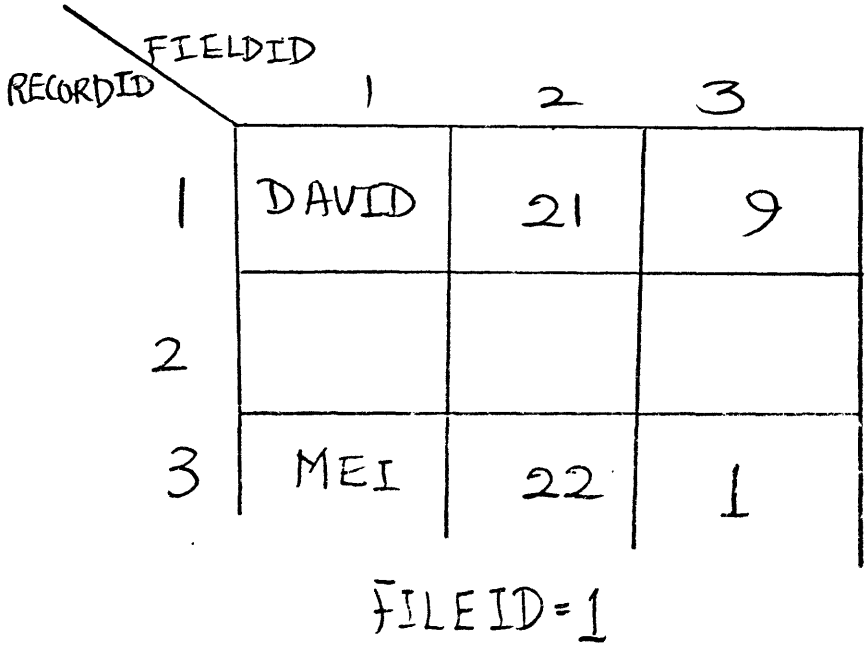
2.2 U-nary Concepts

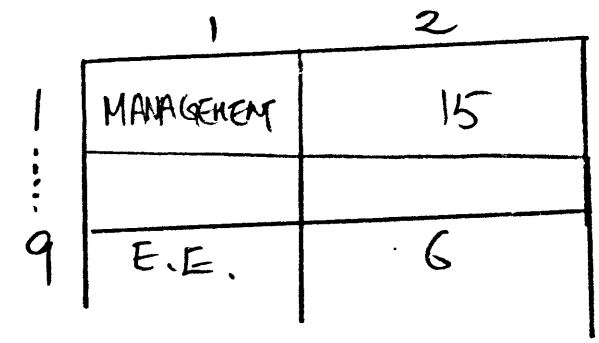
At the U-nary Level, the data are organized in files, records and fields. In principal, a Pset would be mapped to a file. A file is divided into records and fields. A record is

contain the actual data. Each file has a unique numerical ID. Within each file, each record has a unique recordid. Within each record, each field has a unique fieldid. Each field is designated to contain a pre-determined kind of data.

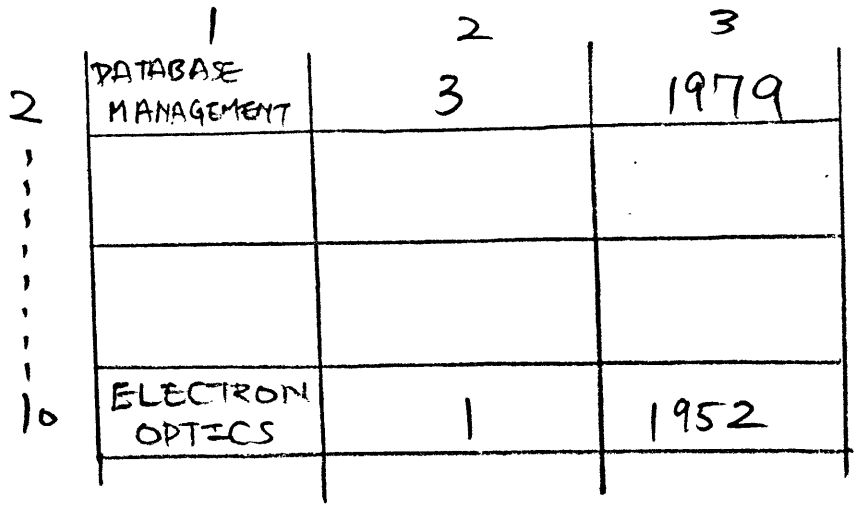
There are two kinds of data- actual numerical or character data, and recordid. In order to implement relation or Bset in the U-nary Level, recordids are used. If the relation between two Psets is 'E', then the recordid of the child Pset is included in a designated data field of the record of the parent to mimic the relationship.

To continue with the two examples presented earlier in the discussion of data organization in the Entity Level, here are the records that correspond to the information contained in the entity level:





FILEID=8



FILEID= 6

The crux of the problem lies in how to map the data in the Entity Level to the data in the U-nary Level. The work is done by the N-ary level and part of the work is accomplished by my thesis.

Chapter Three

Functional Modules, Data Structures and Strategy

3.1 Examplary Update trees

Although all the three operations use the same data structures, they differ subtly in their contents. A discussion of an example for each is obviously warranted. As a preview, a discussion of the general structure of an update tree would precede that of any specific tree.

All the information that is required for a certain kind of update, whether Create, Delete or Modify are formulated by the level above in an update tree. The update tree contains five kinds of nodes: one rootnode, value nodes, many-to-1(mto1) entity nodes, one-to-one(1to1) entity nodes and one-to-many(1tom) entity nodes. The rootnode is the file which contains the record to be updated. All the attached nodes are either of two kinds- value or entity. A value node may contain only one data value , e.g., age. An entity node is another record. For example, a tree that models Mei's personal record to be updated is as follows:

Mei is of age 22, home state being Ma, whose job is student, office address being Bldg 53 and of salary $2000/mthly. The tree that contains all this information is as follows:

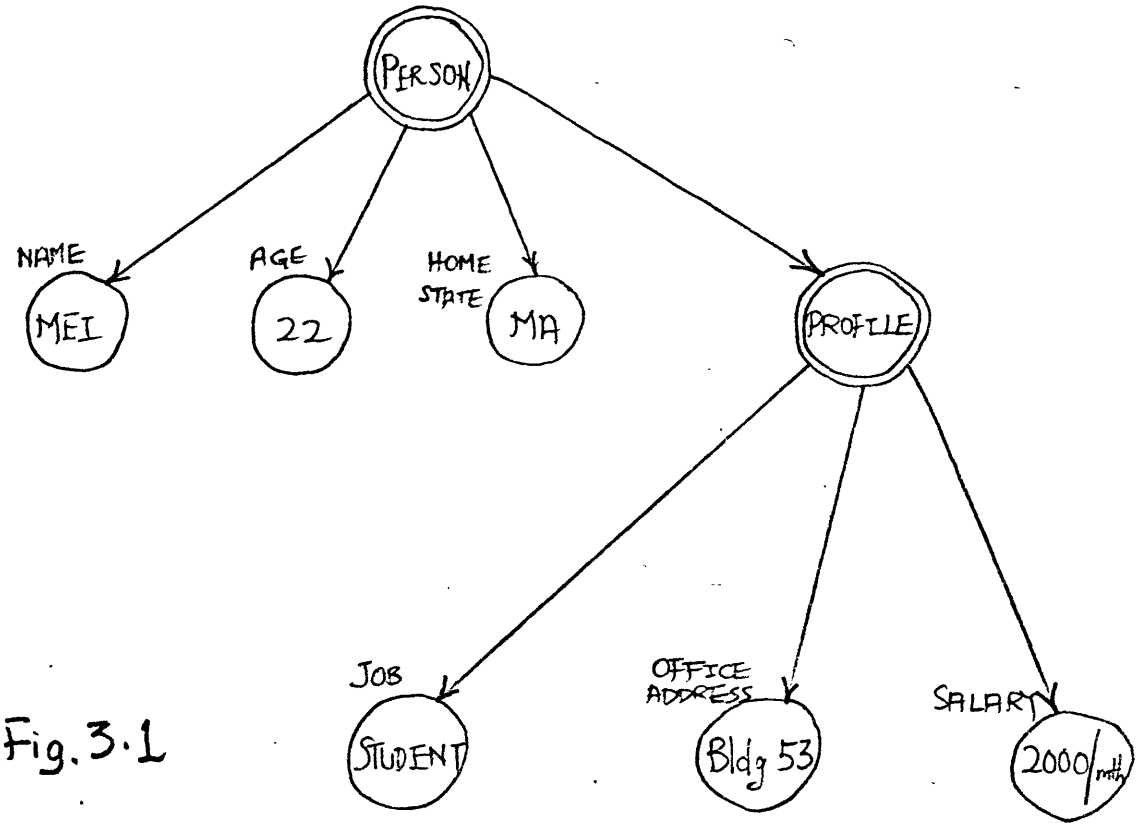


Fig. 3.1

Basically, there are three kinds of operations involved: create a new record in a certain file and, if necessary, insert this new recordid in existing record; delete an existent record; modify the value contents or recordid contents of an existing record.

For creation, three kinds of nodes are attached to the rootnode. All the values or recordid in these nodes are to be

filled in the new record to be created or if the new record is supposed to be contained in one of these nodes, the recordid of this newly created node is inserted into the appropriate record. The value nodes have their values contained in the UPDN_ARG which is the control block between the N-ary level and the previous level. The entity nodes have sufficient nodes attached to them so as to enable this level to retrieve the recordid.

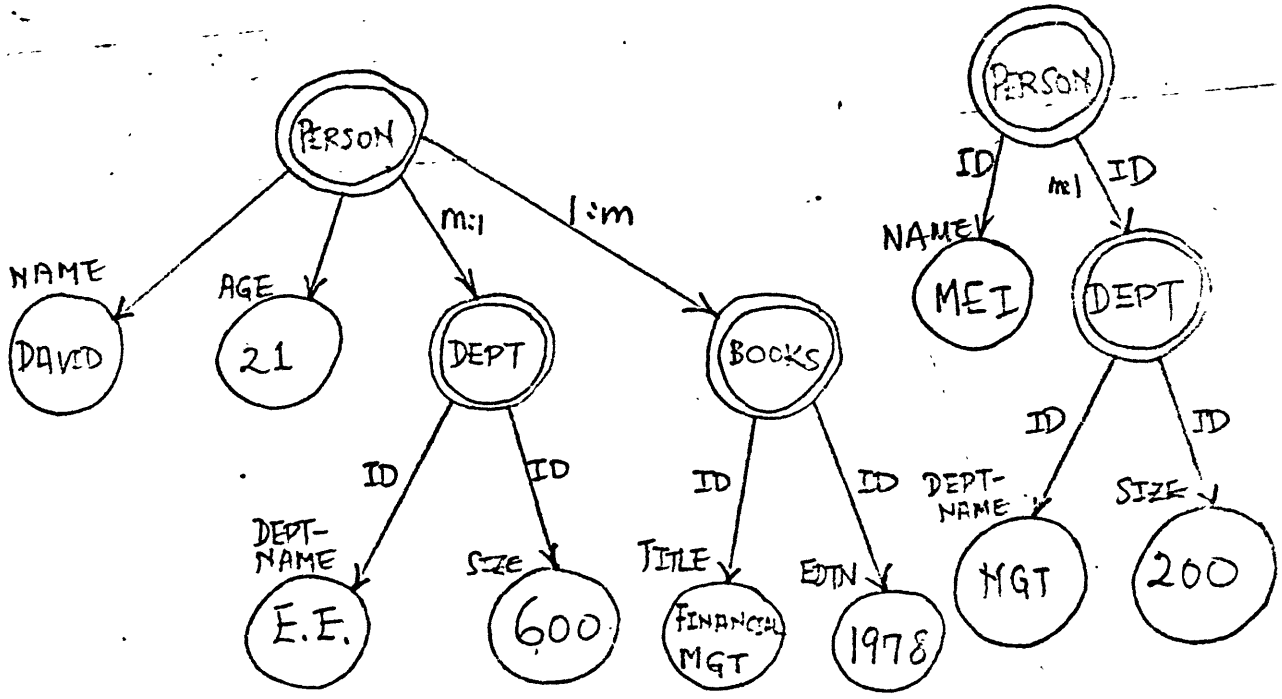


Fig. 3.2

For deletion, all the nodes that are attached to rootnode, whether it is a value or entity node, are contained within the rootnode to identify the record. The recordid retrieved is passed down to the next level for deletion.

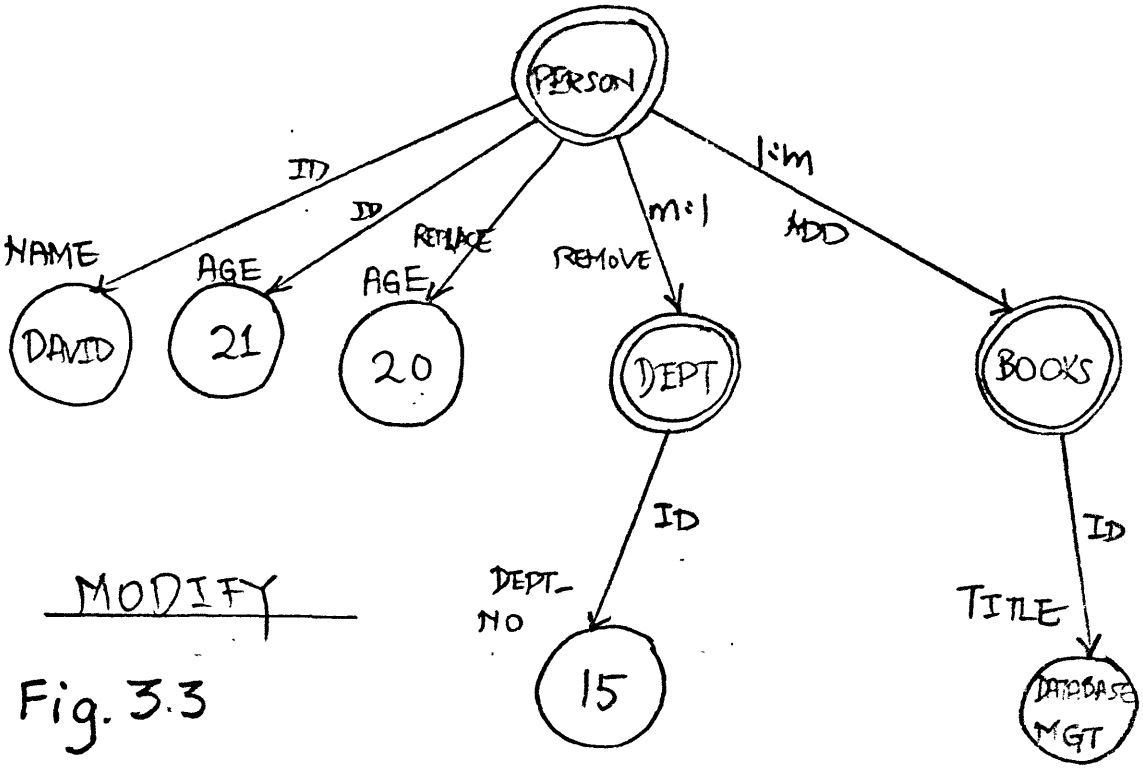


Fig. 3.3

For modification, there are four kinds nodes as in creation: the value nodes, the mtol entity nodes, the ltol entity nodes, and the ltom entity nodes. The three kinds of operations are ID, REPLACE, ADD, REMOVE. The ID is to identify the record to be modified. The REPLACE changes the value contents of the value nodes. ADD and REMOVE simply add or remove a Bset relation.

3.2 Modules of N-ary Planned Implementation

As a preliminary consideration, the implementation consists of a main program which takes care of the control and calling of subroutines. The main program would be separated into two parts. The first part deals with mtol and embedded

1to1 relationship only while the second part deals with non-embedded 1to1 relationship and 1tom relationship. It is useful to separate one from the other because in solving some non-embedded relationship issues it is necessary to obtain information from the embedded relationship. Also it is a good idea to maintain a clear-cut distinction between the two because the coding would have very little similarity and hence, it is not meaningful to merge them together.

Within the upper half of the main program, there is a further distinction between the part that deals with value nodes and the part that deals with entity nodes, all nodes being mto1 or embedded 1to1 relation only. The reason for this separation is again the difference in implementation outweighs similarity in spite of a lot of conceptual similarity.

Information about the two kinds of nodes are extracted from the UPDN_ARG received from the upper level. For value nodes, the fieldid is to be retrieved from the catalogue and fro entity nodes, the recordid is to be retrieved with the help of the retrieval subsystem. So there would be quite a number of interfaces with the retrieval subsystem and the catalogue.

The second half deals with non-embedded 1to1 and 1tom relationship. Combined with information obtained from the first stage, it would formulate a request for each non-embedded

relationship. As a result, there would be an array of pointers each pointing to a control block forming a request.

At preliminary glance, there would be about 20-25 subroutines. They are grouped into modules calling each other in the same level. The following is a crude flowchart of calling modules.

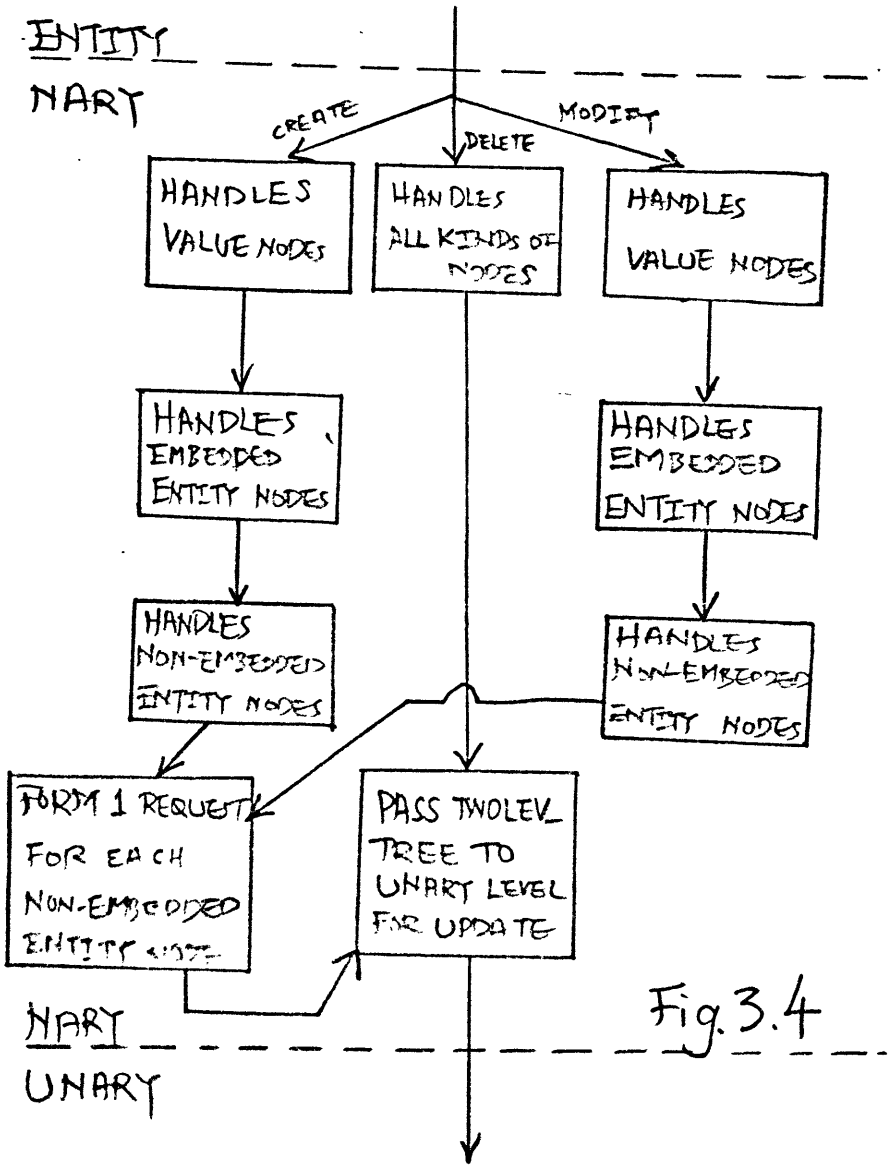


Fig. 3.4

3.3 Data Structures

Before an explanation of the strategy is discussed, it is useful to review what is submitted to the N-ary Level. The structure is called UPDN_ARG and it is declared as follows:

```
dcl 1 updn_arg(p),
        2 basic_op char(1), /* 'c', 'd', or 'm' */
        2 root_psetid bit(32),
            /* unique psetid to be mapped to a unique file
        2 n fixed bin, /* total number of non-root nodes
        2 node_descrip(ua_ctr refer(updn_arg.n))
            3 bsetid bit(32),
                /* unique bsetid to be mapped to a field
            3 parent fixed bin, /* node no of parent */
            3 op char(1), /* meaningful only for modify
                             'p', 'r', or 'a' */
                3 data char(40); /* numerical, character
                                    or recordid */
```

Each node in the tree is numbered with differently. The whole tree in the entity level is mapped into this structure before it is submitted to the N-ary Level for further mapping. The rootnode is identified with a root psetid. All the other nodes are represented as an element of an array of node_descrip. Again, each node is identified with its unique

bsetid, its parent number in the tree, its operation code, if any, and the data if it is a value node.

What is returned to the N-ary Level is the UPDN_RTN with the following declaration:

```
dcl 1 updn_rtn based(p),
        2 rtn_code fixed bin(15), /* 0= success, */
        2 error_nodeno fixed;
            /* node number with invalid recordid */
```

The rtn_code is indicative of the success or failure of the basic operation, create, delete or modify after the retrieval of all the recordids are successfully completed. The error_nodeno is indicative of which recordid cannot be retrieved. It is the node number of the entity node of this subtree whose recordid is invalid.

The structure that is submitted to the U-nary Level after all the mappings are successfully completed is as follows:

```
dcl 1 twolev_tree based(p),
        2 basic_op char(1), /* 'c', 'd' or 'm' */
        2 b_op_rtncode fixed bin,
            /* to be filled by U-nary Level */
        2 root_fileid fixed bin, /* fileid of root node
        2 recordid fixed bin, /* recordid of root node
```

```
    2 no_of_field, /* total no of fields */
    2 id_arr(no_fdupd refer(twolev_tree.no_of_field),
        3 fieldid fixed bin(15),
        3 data char(40); /* actual data or recordid
```

The root_fileid identifies the file whose contents are to be updated. Each element of the id_arr indicates the particular field to be updated and the data to be used to replace the current content.

The structure that is submitted back to the N-ary Level from the U-nary Level is the same one that is passed down. However, there is a new piece of information- the b_op_rtncode which is filled by the U-nary Level as according to whether the basic operation is successful or not.

Finally, an important structure needs to be mentioned is the catalogue where all the mappings information are extracted. It is able to map a psetid to a fileidand a bsetid to a fieldid. Also information as to the type of the bset relation, e.g., 'E' or 'N' for 1to1 relation, and finally the fileid of the target node for a non-embedded 1to1 relation are obtained.

```
dcl 1 pcat based(p),
        2 psetid bit(32),
        2 fileid fixed bin(15),
```

```
2 n fixed bin, /* no of assoc. of this Pset */
2 bset_descrip(ct_ctr refer(pcat.n))
    3 bsetid bit(32),
    3 type char(1), /* 'v' or 'n' */
    3 func char(1), /* 'm', '1' or 'o' */
    3 imp,
        4 itype char(1), /* 'e' or 'n' */
        4 fieldid fixed,
        4 fileid fixed bin(31);
            /* fileid of target node */
```

A few points need be mentioned here. Type means whether the node is a value node or entity node. Func means whether the relation is an mtol, 1to1 or 1tom. Itype means whether or not the data value or recordid is embedded in the current file. Fileid is non-zero, of course, only when the target node is a 1tom entity node or a non-embedded 1to1 entity node. It is obvious from the previous layout of the structure of the catalogue that practically no mappings can be accomplished without access to the information to the appropriate catalogue which corresponds to the root_psetid in the UPDN_ARG.

3.4 Strategy

From the previous discussion, it is obvious that the mapping information comes from extracting the proper catalogue

entry and deducing all the fields to be updated. However, due to the existence of non-embedded entity nodes, i.e., those with the parameter 'N', which imply that the file of the target node needs to be modified rather than that of the source node or the rootnode, it is necessary to construct an individual twolev_tree for each such non-embedded node. The strategy is to break down the update tree into two groups---- (i) embedded nodes: value nodes, mtol entity nodes, embedded ltol entity nodes (ii) non-embedded nodes: non-embedded ltol entity nodes, ltom entity nodes. The reason is obvious- for embedded nodes, all the data, whether it is actual numerical or character value, are contained in the same one file, that of the rootnode. However, for non-embedded nodes, the data are contained in the file of the target nodes. Hence there is more than one file to be updated. However, the twolev_tree can only accommodate all the update information for one file. It is therefore essential to construct one twolev_tree for each file to be modified. To summarize, if the tree has two non-embedded ltol ot ltom entity child nodes, then three twolev_trees are passed down to the U-nary Level for complete updating.

3.5 Preview of Important Functional Modules

As mentioned before, update would consist of create, delete and modify. For create and modify, it is necessary to break down the tree into embedded and non-embedded nodes. The

modules, cr2, cr19, md2, md19 are responsible for constructing the twolev_tree for the root_node. Cr20 and md20 are responsible for constructing a twolev_tree for each non-embedded entity node. For delete, only the record of one file needs to be deleted. Hence, only de2 and del9 exist. A discussion of each functional module is presented in the following chapter.

Chapter Four

Description of Individual Functional Modules

4.1 Top Level Modules

The main program directs operation to cr2, cr19 etc., as according to the basic_op. Top level procedures are those that are visible in the main program. However, an important second level subroutine must be mentioned prior to the discussion of top level procedures, cr2, cr19, cr20, de2, del9, md2, md19 and md20. The subroutine is GNI.

GNI- in principal, it extracts all the information about each node from the catalogue and the UPDN_ARG. For value nodes, it stores the fieldid and data in a structure pointed to by MVNIP; for embedded entity nodes, it stores the first node of the sub-tree and the fieldid of the field that contains the recordid in a structure pointed to by MNNIP; for non-embedded entity nodes, it also stores the initial node no in a structure pointed to by OMNIP.

CR2- it invokes GNI to obtain information on value nodes with which it uses to fill the fieldid and data for all the value nodes in the twolev_tree of the rootnode.

CR19- it also employs the information on the initial node number of embedded entity nodes obtained from the CR2 invocation of GNI. It deals primarily with embedded entity nodes. It locates the number of the last node in the subtree. Then it passes the whole subtree to the retrieval subsystem to obtain the recordid defined by this subtree. It then fills the data part of the twolev_tree with this recordid. After all the recordids are retrieved, the twolev_tree for updating the file that corresponds to the rootnode is ready for export to the U-nary Level.

CR20- it also employs the information on the initial node number of non-embedded entity nodes obtained from the CR2 invocation of GNI. It deals primarily with non-embedded nodes. It first finds the fileid of the non-embedded node from the catalogue and then generate a twolev_tree for that non-embedded node. It is obvious that this twolev_tree has only one node and the data is the recordid of the newly created record.

DE2- it maps the root_psetid to the proper fileid.

DE19- it passes the whole tree to the retrieval subsystem and then inserts the returned recordid into the twolev_tree. MD2-

essentially the same as CR2 except that not all fields are updated. In CR2, if a certain field is not specified in the UPDN_ARG, it is still created with an initial value of NULL. However, in MD2, only specified fields are updated. MD19, MD20- same as their CREATE counterparts except that it is essential to fill the twolev_tree with the information on nodes that are to be removed before those that are to be added.

ZGTCT- it extracts the appropriate catalogue that corresponds to the given root_psetid in the UPDN_ARG.

TPROC- it is responsible for submitting a twolev_tree to the U-nary Level every time it is invoked. Hence it is an indispensable interface module.

CONTROL- this is the main program monitoring the flow of control. It directs execution to the proper create, delete or modify modules. It is also responsible for interfacing with the retrieval subsystem, the Entity Level and the U-nary Level.

USER- this is the options main program which enables the user to test all the modules with user-supplied data.

4.2 Second Level Subroutines

The aforementioned top level procedures call other subroutines that do not appear in the main control program. Hence they are called Second Level procedures. They can be separated into two main kinds- those that query the extracted catalogue entry and those that do not. This section deals with those that do not.

ILN- this boolean procedure test if a node is a second level node in an update tree, UPDN_ARG.

CLN- it counts the total number of second level nodes.

CMN- it counts the total number of second level value nodes and the total number of second level embedded nodes.

GTRCID- it is responsible for interfacing with the retrieval subsystem for extracting the recordid of embedded subtrees. It does so by passing the whole subtree to the retrieval subsystem.

4.3 Query Catalogue Subroutines

The following subroutines are mainly responsible for carrying out all the mapping functions. They are therefore very important and act as the bridge between the Entity Level and the U-nary Level.

QCEMB- this boolean procedure checks whether a given node is of type embedded, 'E', or non-embedded, 'N'.

QCFDID- this procedure returns the fieldid of a given node. If it is an embedded node, it returns the field in the record of the file of the rootnode that is to be updated. If it is a non-embedded node, it returns the field in the record of the file of the target that is to be updated.

QCFLDS- this procedure returns the fileid of a target node. Hence it is invoked only when the calling procedure deals with non-embedded nodes.

QCFLID- this procedure returns the fileid of the root node.

QCNTYP- this procedure returns the type of the node. It can be value node, 'V', or an entity node, 'N'.

4.4 Conclusion

All the procedures described above are actually to be integrated into the main program of the STV to consitute the N-ary Level. In the following and last chapter, we will discuss the simulation of the environment so that the above procedures can be deterministically tested against programming bugs.

Hence the procedures that will be mentioned are not part of the N-ary Level but merely written to ensure binding testings are carried out.

Chapter Five

Simulation and Simulated Procedures

5.1 Concepts

There is a fine distinction between simulation procedures and simulated procedures. Because of the incompletion of the whole N-ary Level, some procedures that ought to be present to enable full testing of the updat subsystem will have to be simulated. However, some structures like the database should never ought be created by the N-ary Level. Nevertheless, they are coded to enable the simulation of the actual environment to the fullest extent. These are simulation procedures. Simulated procedures that ought to be in the N-ary Level have been discussed in the previous chapter. These include GTRCID, TPROC and ZGTCT. Simulation procedures are discussed in the following section.

5.2 Simulation Subroutines

Simulation procedures include SBDUA and SFLAPC. The former is responsible for interacting with the user to receive

user-supplied update trees, UPDN_ARG. The latter is responsible for building up a mini database.

SBDUA- It fires requests to the user for all the necessary information to build up a full-bloom update tree. In order, it queries the total no of non-root nodes, the basic operation, the root_psetid, for each non-root node, the bsetid, the parent, the operation code and finally the data.

SFLAPC- It is a predetermined procedure in the sense that when the user calls the main program CONTROL, it merely creates the preassigned database. The whole database is as depicted pictorally at the end of this chapter.

5.3 Conclusion

With the presence of these simulated and simulation procedures, the user can fully test the update subsystem. A sample terminal session is provided in appendix three. All the user-supplied interfaces have been coded into exec files for simplicity. Notice the large number of exec files tested. Care has been taken so that each exec file tests for different characteristics of the subsystem and overlap has been avoided to the best knowledge of the author. The exec files from 30 onwards test the modules that handle update trees with CREATE as the basic_op. The exec files from 50 onwards test the mod-

ules that handle update trees with DELETE as the basic_op. The exec files from 70 onwards test the modules that handle update trees with MODIFY as the basic_op.

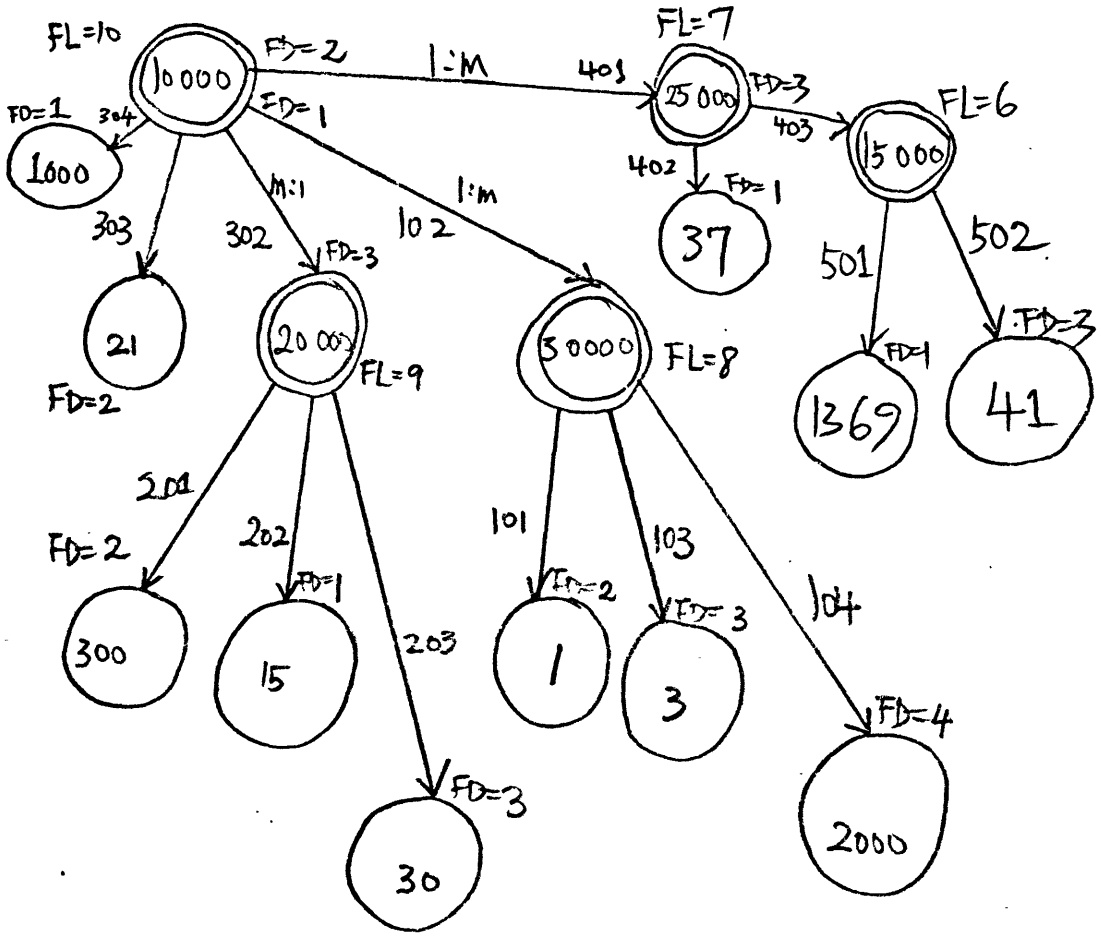


Fig. 5.1

Bibliography

1. Chat-Yu Lam, Stuart E. Madnick
   'INFOPLEX Data Base Computer Architecture'
   Center for Information Systems Research, M.I.T.

2. Meichun Hsu
   'FSTV - The Software Test Vehicle for the Functional Hierarchy of the INFOPLEX Data Base Computer'
   Center for Information Systems Research, M.I.T.

3. C. J. Date
   'An Introduction to Database Management' Second Edition
   Addison-Wesley.

4. Bruce Blumberg
   'INFOSAM - A Sample Database Management System'
   Center for Information Systems Research, M.I.T.

5. Meichun Hsu
   'Draft of the FSTV'
   Center for Information Systems Research, M.I.T.

Appendix 1- Program Listings

The following P/L1 program modules are included in this appendix in order.

GNI

CR2

CR19

CR20

DE2

DE19

DE20

MD2

MD19

MD20

ZGTCT

TPROC

CONTROL

USER

ILN

CLN

CMN

GTRCID

QCEMB

QCFDID

QCFLDS

QCFLID

QCNTYP

SBDUA

SFLAPC

```
*PROCESS NAME('GNI'), INCLUDE, F(I);                                    GNI00010
 GNI: PROC (UAP, BASIC_OP, MVNIP, MNNIP, OMNIP,                         GNI00020
           NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE, CP, CT_NO);         GNI00030
                                                                        GNI00040
                                                                        GNI00050
 /* GNI= GET_NODEINFO */                                                GNI00060
                                                                        GNI00070
                                                                        GNI00080
      %INCLUDE MVNI;                                                    GNI00090
      %INCLUDE MNNI;                                                    GNI00100
      %INCLUDE ONI;                                                     GNI00110
      %INCLUDE UA;                                                      GNI00120
                                                                        GNI00130
                                                                        GNI00140
      DCL P PTR;                                                        GNI00150
      DCL ILN ENTRY EXTERNAL RETURNS(BIT),                              GNI00160
          QCNTYP ENTRY EXTERNAL RETURNS(CHAR(1)),                       GNI00170
          QCFDID ENTRY EXTERNAL RETURNS(FIXED BIN(31)),                 GNI00180
          QCEMB ENTRY EXTERNAL RETURNS(BIT);                            GNI00190
                                                                        GNI00200
                                                                        GNI00210
      DCL UAP PTR,                                                      GNI00220
          BASIC_OP CHAR(1),                                             GNI00230
          (MVNIP, MNNIP, OMNIP) PTR,                                    GNI00240
          (NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE) FIXED,              GNI00250
          CP PTR,                                                       GNI00260
          CT_NO FIXED;                                                  GNI00270
                                                                        GNI00280
                                                                        GNI00290
      DCL (VIA_CTR, NIA_CTR, ONIA_CTR, NODE_NO) FIXED INIT(0),          GNI00300
          BSETID FIXED BIN(15) INIT(0);                                 GNI00310
                                                                        GNI00320
                                                                        GNI00330
      VIA_CTR= NO_MTO1VNODE;                                            GNI00340
      NIA_CTR= NO_MTO1NNODE;                                            GNI00350
      ONIA_CTR= NO_1TOMNODE;                                            GNI00360
      IF NO_MTO1VNODE^= 0                                               GNI00370
      THEN                                                              GNI00380
      DO;                                                               GNI00390
      ALLOCATE MTO1VNODE_INFO SET (MVNIP);                              GNI00400
      MVNIP-> MTO1VNODE_INFO.INFO_ARR(*).NODE_NO= 0;                    GNI00410
      MVNIP-> MTO1VNODE_INFO.INFO_ARR(*).FIELDID= 0;                    GNI00420
      MVNIP-> MTO1VNODE_INFO.INFO_ARR(*).DATA= ' ';                     GNI00430
      END;                                                              GNI00440
      IF NO_MTO1NNODE^= 0                                               GNI00450
      THEN                                                              GNI00460
      DO;                                                               GNI00470
      ALLOCATE MTO1NNODE_INFO SET (MNNIP);                              GNI00480
      MNNIP-> MTO1NNODE_INFO.INFO_ARR(*).INITIAL_NODE_NO= 0;            GNI00490
      MNNIP-> MTO1NNODE_INFO.INFO_ARR(*).FINAL_NODE_NO= 0;              GNI00500
      MNNIP-> MTO1NNODE_INFO.INFO_ARR(*).FIELDID= 0;                    GNI00510
      END;                                                              GNI00520
      IF NO_1TOMNODE^= 0                                                GNI00530
      THEN                                                              GNI00540
      DO;                                                               GNI00550
```

```
      ALLOCATE OTOMNODE_INFO SET (OMNIP);                                    GNI00560
      OMNIP-> OTOMNODE_INFO.INFO_ARR(*)= 0;                                  GNI00570
      END;                                                                   GNI00580
      VIA_CTR= 0;                                                            GNI00590
      NIA_CTR= 0;                                                            GNI00600
      ONIA_CTR= 0;                                                           GNI00610
                                                                             GNI00620
      DO NODE_NO= 1 TO UAP-> UPDN_ARG.N;                                     GNI00630
       IF ILN(UAP, NODE_NO)                                                  GNI00640
        THEN                                                                 GNI00650
         DO;                                                                 GNI00660
                  IF BASIC_OP= 'M'                                           GNI00670
                     & UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO)                  GNI00680
                              .OP= 'I'                                       GNI00690
                     THEN GOTO EOL;                                          GNI00700
            BSETID= UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).BSETID;             GNI00710
            IF QCNTYP(CP, CT_NO, BSETID)= 'V'                                GNI00720
               THEN     DO;                                                  GNI00730
                           VIA_CTR= VIA_CTR + 1;                             GNI00740
                           MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).         GNI00750
                              NODE_NO= NODE_NO;                              GNI00760
                           MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).         GNI00770
                              FIELDID= QCFDID(CP, CT_NO, BSETID);            GNI00780
                           MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).         GNI00790
                              DATA= UAP-> UPDN_ARG.NODE_DESCRIP              GNI00800
                                       (NODE_NO).DATA;                       GNI00810
                         END;                                                GNI00820
               ELSE                                                          GNI00830
                IF QCEMB(CP, CT_NO, BSETID)                                  GNI00840
                   THEN                                                      GNI00850
                    DO;                                                      GNI00860
                        NIA_CTR= NIA_CTR + 1;                                GNI00870
                        MNNIP-> MTO1NNODE_INFO.INFO_ARR                      GNI00880
                           (NIA_CTR).INITIAL_NODE_NO= NODE_NO;               GNI00890
                        MNNIP-> MTO1NNODE_INFO.INFO_ARR                      GNI00900
                           (NIA_CTR).FIELDID                                 GNI00910
                           = QCFDID(CP, CT_NO, BSETID);                      GNI00920
                    END;                                                     GNI00930
                   ELSE                                                      GNI00940
                    DO;                                                      GNI00950
                     ONIA_CTR= ONIA_CTR + 1;                                 GNI00960
                     OMNIP-> OTOMNODE_INFO.INFO_ARR(ONIA_CTR)= NODE_NO;      GNI00970
                    END;                                                     GNI00980
         END;                                                                GNI00990
EOL:                                                                         GNI01000
     END;                                                                    GNI01010
                                                                             GNI01020
                                                                             GNI01030
END GNI;                                                                     GNI01040
```

```
*PROCESS NAME('CR2'), INCLUDE, F(I);                                    CR200010
 CR2: PROC (UAP, BASIC_OP, MVNIP, MNNIP, OMNIP,                         CR200020
             PTLTP, NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE,            CR200030
             CP, CT_CTR);                                               CR200040
                                                                        CR200050
                                                                        CR200060
 /* CR2= CREATE2 */                                                     CR200070
                                                                        CR200080
                                                                        CR200090
      %INCLUDE TWOLEV;                                                  CR200100
      %INCLUDE MVNI;                                                    CR200110
      %INCLUDE UA;                                                      CR200120
      %INCLUDE SPCT;                                                    CR200130
                                                                        CR200140
                                                                        CR200150
      DCL P PTR;                                                        CR200160
      DCL CLN ENTRY EXTERNAL RETURNS(FIXED),                            CR200170
          CMN ENTRY EXTERNAL,                                           CR200180
          QCFLID ENTRY EXTERNAL RETURNS(FIXED BIN(31)),                 CR200190
          GNI ENTRY EXTERNAL,                                           CR200200
          ILN ENTRY EXTERNAL RETURNS(BIT);                              CR200210
                                                                        CR200220
                                                                        CR200230
      DCL UAP PTR,                                                      CR200240
          BASIC_OP CHAR(1),                                             CR200250
          (MVNIP, MNNIP, OMNIP, PTLTP) PTR,                             CR200260
          (NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE) FIXED,              CR200270
          CP PTR,                                                       CR200280
          CT_CTR FIXED;                                                 CR200290
                                                                        CR200300
                                                                        CR200310
      DCL (NO_LEVEL2NODE, NO_FDUPD, VIA_CTR, ND_CTR, FD_CTR,            CR200320
              IDARR_CTR) FIXED INIT(0);                                 CR200330
                                                                        CR200340
                                                                        CR200350
      NO_LEVEL2NODE= CLN(UAP);                                          CR200360
      CALL CMN(UAP, BASIC_OP, NO_MTO1VNODE, NO_MTO1NNODE, CP, CT_CTR);  CR200370
      DO FD_CTR= 1 TO CP-> PCAT(CT_CTR).N;                              CR200380
         IF CP-> PCAT(CT_CTR).BSET_DESCRIP(FD_CTR).IMP.ITYPE= 'E'       CR200390
            THEN NO_FDUPD= NO_FDUPD + 1;                                CR200400
      END;                                                              CR200410
                                                                        CR200420
      NO_1TOMNODE= NO_LEVEL2NODE - NO_MTO1VNODE - NO_MTO1NNODE;         CR200430
                                                                        CR200440
      ALLOCATE TWOLEV_TREE SET (PTLTP);                                 CR200450
      PTLTP-> TWOLEV_TREE.BASIC_OP= ' ';                                CR200460
      PTLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;                              CR200470
      PTLTP-> TWOLEV_TREE.ROOT_FILEID= 0;                               CR200480
      PTLTP-> TWOLEV_TREE.RECORDID= 0;                                  CR200490
      PTLTP-> TWOLEV_TREE.ID_ARR(*).FIELDID= 0;                         CR200500
      PTLTP-> TWOLEV_TREE.ID_ARR(*).DATA= ' ';                          CR200510
                                                                        CR200520
      PTLTP-> TWOLEV_TREE.BASIC_OP= BASIC_OP;                           CR200530
      PTLTP-> TWOLEV_TREE.ROOT_FILEID= QCFLID(CP, CT_CTR);              CR200540
      DO FD_CTR= 1 TO CP-> PCAT(CT_CTR).N;                              CR200550
```

1-86-1

```
        IF CP-> PCAT(CT_CTR).BSET_DESCRIP(FD_CTR).IMP.ITYPE= 'E'        CR200560
           THEN DO; IDARR_CTR= IDARR_CTR + 1;                           CR200570
                PTLTP-> TWOLEV_TREE.ID_ARR(IDARR_CTR).FIELDID           CR200580
                   = CP-> PCAT(CT_CTR).BSET_DESCRIP(IDARR_CTR).         CR200590
                      IMP.FIELDID;                                      CR200600
                PTLTP-> TWOLEV_TREE.ID_ARR(IDARR_CTR).DATA              CR200610
                   = 'NULL';                                            CR200620
                END;                                                    CR200630
     END;                                                               CR200640
     CALL GNI(UAP, BASIC_OP, MVNIP, MNNIP, OMNIP,                       CR200650
              NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE, CP, CT_CTR);     CR200660
     DO VIA_CTR= 1 TO NO_MTO1VNODE;                                     CR200670
        DO FD_CTR= 1 TO PTLTP-> TWOLEV_TREE.NO_OF_FIELD;                CR200680
     IF PTLTP-> TWOLEV_TREE.ID_ARR(FD_CTR).FIELDID                      CR200690
           = MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).FIELDID           CR200700
   THEN PTLTP-> TWOLEV_TREE.ID_ARR(FD_CTR).DATA                         CR200710
           = MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).DATA;             CR200720
     END;                                                               CR200730
     END;                                                               CR200740
/*   FREE MVNIP-> MTO1VNODE_INFO  */                                    CR200750
                                                                        CR200760
                                                                        CR200770
END CR2;                                                                CR200780
```

```
*PROCESS NAME('CR19'), INCLUDE, F(I);                                   CR100010
 CR19: PROC (UAP, MNNIP, PTLTP, ERR_NODE, CP, FP, CT_NO);               CR100020
 /* CR19= CREATE19 */                                                   CR100030
 /* GTRCID CHECKS FP-> FILE(10).RECORD(2).FILE(3) FOR VALID             CR100040
    RECORDID, HENCE TO TEST FOR INVALID RECORDID, ONE CAN               CR100050
    CHANGE THIS FIELD ENTRY                                             CR100060
    ONE HAS TO TEST FOR TWO CASES                                       CR100070
    1) RECORDID VALID                                                   CR100080
    2) RECORDID INVALID  */                                             CR100090
                                                                        CR100100
                                                                        CR100110
                                                                        CR100120
                                                                        CR100130
      %INCLUDE MNNI;                                                    CR100140
      %INCLUDE UA;                                                      CR100150
      %INCLUDE TWOLEV;                                                  CR100160
                                                                        CR100170
                                                                        CR100180
      DCL ILN ENTRY EXTERNAL RETURNS(BIT);                              CR100190
      DCL GTRCID ENTRY EXTERNAL RETURNS(FIXED);                         CR100200
                                                                        CR100210
                                                                        CR100220
      DCL P PTR;                                                        CR100230
      DCL UAP PTR,                                                      CR100240
          (MNNIP, PTLTP) PTR,                                           CR100250
          (ERR_NODE) FIXED,                                             CR100260
          (CP, FP) PTR,                                                 CR100270
          CT_NO FIXED;                                                  CR100280
                                                                        CR100290
                                                                        CR100300
      DCL (INITIAL_NODE_NO, RTNTND_CTR, FINAL_NODE_NO, NIA_CTR,         CR100310
             RC_RTNCODE, FD_CTR) FIXED INIT(0),                         CR100320
          RECORDID FIXED BIN(31) INIT(0);                               CR100330
                                                                        CR100340
                                                                        CR100350
      DO NIA_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;             CR100360
         INITIAL_NODE_NO= MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).     CR100370
                          INITIAL_NODE_NO;                              CR100380
         DO RTNTND_CTR= INITIAL_NODE_NO + 1 TO UAP->UPDN_ARG.N;         CR100390
            IF RTNTND_CTR= UAP-> UPDN_ARG.N                             CR100400
               THEN MNNIP->MTO1NNODE_INFO.INFO_ARR(NIA_CTR).            CR100410
                    FINAL_NODE_NO= RTNTND_CTR;                          CR100420
          ELSE IF ILN(UAP, RTNTND_CTR)                                  CR100430
               THEN DO;                                                 CR100440
                       MNNIP->MTO1NNODE_INFO.INFO_ARR(NIA_CTR).         CR100450
                          FINAL_NODE_NO= RTNTND_CTR - 1;                CR100460
                       RTNTND_CTR= UAP->UPDN_ARG.N + 1;                 CR100470
                    END;                                                CR100480
         END;                                                           CR100490
      END;                                                              CR100500
      DO NIA_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;             CR100510
         INITIAL_NODE_NO= MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).     CR100520
                          INITIAL_NODE_NO;                              CR100530
         FINAL_NODE_NO= MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).       CR100540
                          FINAL_NODE_NO;                                CR100550
```

```
        RC_RTNCODE= GTRCID(UAP, INITIAL_NODE_NO, FINAL_NODE_NO,          CR100560
                           RECORDID, CP, FP, CT_NO);                     CR100570
        IF RC_RTNCODE^= 0                                                CR100580
           THEN DO;                                                      CR100590
                   ERR_NODE= INITIAL_NODE_NO;                            CR100600
                   GOTO RETN;                                            CR100610
                END;                                                     CR100620
        DO FD_CTR= 1 TO PTLTP-> TWOLEV_TREE.NO_OF_FIELD;                 CR100630
                                                                         CR100640
      IF PTLTP-> TWOLEV_TREE.ID_ARR(FD_CTR).FIELDID                      CR100650
            = MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).FIELDID           CR100660
    THEN PTLTP-> TWOLEV_TREE.ID_ARR(FD_CTR).DATA                         CR100670
            = RECORDID;                                                  CR100680
        END;                                                             CR100690
      END;                                                               CR100700
RETN:                                                                    CR100710
/*    FREE MNNIP-> MTO1NNODE_INFO */                                     CR100720
                                                                         CR100730
                                                                         CR100740
END CR19;                                                                CR100750
```

```
*PROCESS NAME('CR20'), INCLUDE, F(I);                                   CR200010
 CR20: PROC (UAP, OMNIP, PTLTP, ERR_NODE, PAP, CP, FP, CT_NO);          CR200020
 /* CR20= CREATE20 */                                                   CR200030
 /* 2 CASES TO TEST FOR :                                              CR200040
    1) RECORDID OF THE 1TOMNODE VALID                                   CR200050
    2) RECORDID OF THE 1TOMNODE INVALID */                              CR200060
                                                                        CR200070
                                                                        CR200080
      %INCLUDE UA;                                                      CR200090
      %INCLUDE TWOLEV;                                                  CR200100
      %INCLUDE ONI;                                                     CR200110
      %INCLUDE PA;                                                      CR200120
                                                                        CR200130
                                                                        CR200140
      DCL NULL BUILTIN;                                                 CR200150
      DCL QCFLDS ENTRY EXTERNAL RETURNS(FIXED BIN(31));                 CR200160
      DCL ILN ENTRY EXTERNAL RETURNS(BIT);                              CR200170
      DCL GTRCID ENTRY EXTERNAL RETURNS(FIXED);                         CR200180
      DCL QCFDID ENTRY EXTERNAL RETURNS(FIXED BIN(31));                 CR200190
                                                                        CR200200
                                                                        CR200210
      DCL P PTR;                                                        CR200220
      DCL (UAP, OMNIP, PTLTP) PTR,                                      CR200230
          ERR_NODE FIXED,                                               CR200240
          (PAP, CP, FP) PTR,                                            CR200250
          CT_NO FIXED;                                                  CR200260
                                                                        CR200270
                                                                        CR200280
      DCL BASIC_OP CHAR(1) INIT(' '),                                   CR200290
          NO_FDUPD FIXED INIT(0),                                       CR200300
          ROOT_RECORDID FIXED BIN(31) INIT(0),                          CR200310
          (ONIA_CTR, NO_1TOMNODE, NODE_NO) FIXED INIT(0),               CR200320
          BSETID FIXED BIN(15) INIT(0),                                 CR200330
          STLTP PTR INIT(NULL()),                                       CR200340
          (INITIAL_NODE_NO, RTNTND_CTR, FINAL_NODE_NO, RC_RTNCODE)      CR200350
             FIXED INIT(0),                                             CR200360
          RECORDID FIXED BIN(31) INIT(0),                               CR200370
          PA_CTR FIXED INIT(0);                                         CR200380
                                                                        CR200390
                                                                        CR200400
      BASIC_OP= 'M';                                                    CR200410
      NO_FDUPD= 1;                                                      CR200420
      DO ONIA_CTR= 1 TO OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;              CR200430
         NODE_NO= OMNIP-> OTOMNODE_INFO.INFO_ARR(ONIA_CTR);             CR200440
         BSETID= UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).BSETID;           CR200450
         ALLOCATE TWOLEV_TREE SET (STLTP);                              CR200460
         STLTP-> TWOLEV_TREE.BASIC_OP= ' ';                             CR200470
         STLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;                           CR200480
         STLTP-> TWOLEV_TREE.ROOT_FILEID= 0;                            CR200490
         STLTP-> TWOLEV_TREE.RECORDID= 0;                               CR200500
         STLTP-> TWOLEV_TREE.ID_ARR.FIELDID= 0;                         CR200510
         STLTP-> TWOLEV_TREE.ID_ARR.DATA= ' ';                          CR200520
         STLTP-> TWOLEV_TREE.BASIC_OP= BASIC_OP;                        CR200530
         STLTP-> TWOLEV_TREE.ROOT_FILEID= QCFLDS(CP, CT_NO, BSETID);    CR200540
         INITIAL_NODE_NO= NODE_NO;                                      CR200550
```

```
        DO RTNTND_CTR= INITIAL_NODE_NO + 1 TO UAP->UPDN_ARG.N;          CR200560
           IF RTNTND_CTR= UAP-> UPDN_ARG.N                              CR200570
              THEN FINAL_NODE_NO= RTNTND_CTR;                           CR200580
      ELSE IF ILN(UAP, RTNTND_CTR)                                      CR200590
              THEN DO;                                                  CR200600
                      FINAL_NODE_NO= RTNTND_CTR - 1;                    CR200610
                      RTNTND_CTR= UAP-> UPDN_ARG.N + 1;                 CR200620
                   END;                                                 CR200630
        END;                                                            CR200640
        RC_RTNCODE= GTRCID(UAP, INITIAL_NODE_NO, FINAL_NODE_NO,         CR200650
                           RECORDID, CP, FP, CT_NO);                    CR200660
        IF RC_RTNCODE^= 0                                               CR200670
           THEN DO;                                                     CR200680
                   ERR_NODE= INITIAL_NODE_NO;                           CR200690
                   GOTO RETN;                                           CR200700
                END;                                                    CR200710
        STLTP-> TWOLEV_TREE.RECORDID= RECORDID;                         CR200720
        STLTP-> TWOLEV_TREE.ID_ARR(1).FIELDID= QCFDID(CP, CT_NO,        CR200730
                                                      BSETID);          CR200740
        PA_CTR= PA_CTR + 1;                                             CR200750
        PAP-> PTR_ARR.ARR(PA_CTR)= STLTP;                               CR200760
     END;                                                               CR200770
RETN:                                                                   CR200780
/*   FREE OMNIP-> OTOMNODE_INFO */                                      CR200790
                                                                        CR200800
                                                                        CR200810
                                                                        CR200820
END CR2O;                                                               CR200830
```

```
*PROCESS NAME('DE2'), INCLUDE, F(I);                                    DE200010
 DE2: PROC (BASIC_OP, PTLTP, CP, CT_CTR);                               DE200020
                                                                        DE200030
                                                                        DE200040
 /* DE2= DELETE2 */                                                     DE200050
                                                                        DE200060
                                                                        DE200070
      %INCLUDE TWOLEV;                                                  DE200080
                                                                        DE200090
                                                                        DE200100
      DCL QCFLID ENTRY EXTERNAL RETURNS(FIXED BIN(31));                 DE200110
                                                                        DE200120
                                                                        DE200130
      DCL P PTR;                                                        DE200140
      DCL BASIC_OP CHAR(1),                                             DE200150
          (PTLTP, CP) PTR,                                              DE200160
          CT_CTR FIXED;                                                 DE200170
      DCL NO_FDUPD FIXED INIT(0);                                       DE200180
                                                                        DE200190
                                                                        DE200200
      NO_FDUPD= 1;                                                      DE200210
      ALLOCATE TWOLEV_TREE SET (PTLTP);                                 DE200220
      PTLTP-> TWOLEV_TREE.BASIC_OP= ' ';                                DE200230
      PTLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;                              DE200240
      PTLTP-> TWOLEV_TREE.ROOT_FILEID= 0;                               DE200250
      PTLTP-> TWOLEV_TREE.RECORDID= 0;                                  DE200260
      PTLTP-> TWOLEV_TREE.ID_ARR(*).FIELDID= 0;                         DE200270
      PTLTP-> TWOLEV_TREE.ID_ARR(*).DATA= ' ';                          DE200280
                                                                        DE200290
      PTLTP-> TWOLEV_TREE.BASIC_OP= BASIC_OP;                           DE200300
      PTLTP-> TWOLEV_TREE.ROOT_FILEID= QCFLID(CP, CT_CTR);              DE200310
                                                                        DE200320
                                                                        DE200330
 END DE2;                                                               DE200340
```

```
*PROCESS NAME('DE19'), INCLUDE, F(I);                                  DE100010
 DE19: PROC (UAP, PTLTP, ERR_NODE, CP, FP, CT_NO);                     DE100020
 /* DE19= DELETE19 */                                                  DE100030
 /* AS A RESULT OF SETTING INITIAL_NODE_NO= 1, GTRCID CHECKS           DE100040
    FP-> FILE(9).RECORD(2).FIELD(2) FOR A VALID RECORD NO              DE100050
    ONE HAS TO TEST FOR TWO CASES                                      DE100060
    1) RECORDID FOR ROOT_NODE VALID                                    DE100070
    2) RECORDID FOR ROOT_NODE INVALID    */                            DE100080
                                                                       DE100090
                                                                       DE100100
                                                                       DE100110
                                                                       DE100120
      %INCLUDE UA;                                                     DE100130
      %INCLUDE TWOLEV;                                                 DE100140
                                                                       DE100150
                                                                       DE100160
      DCL P PTR;                                                       DE100170
      DCL GTRCID ENTRY EXTERNAL RETURNS(FIXED);                        DE100180
                                                                       DE100190
      DCL (UAP, PTLTP) PTR,                                            DE100200
          ERR_NODE FIXED,                                              DE100210
          (CP, FP) PTR,                                                DE100220
          CT_NO FIXED;                                                 DE100230
                                                                       DE100240
                                                                       DE100250
      DCL (INITIAL_NODE_NO, FINAL_NODE_NO, RC_RTNCODE)                 DE100260
             FIXED INIT(0),                                            DE100270
          RECORDID FIXED BIN(31) INIT(0);                              DE100280
                                                                       DE100290
                                                                       DE100300
      INITIAL_NODE_NO= 1;                                              DE100310
      FINAL_NODE_NO= UAP-> UPDN_ARG.N;                                 DE100320
      RC_RTNCODE= GTRCID(UAP, INITIAL_NODE_NO, FINAL_NODE_NO,          DE100330
                         RECORDID, CP, FP, CT_NO);                     DE100340
      IF RC_RTNCODE^= 0                                                DE100350
         THEN DO;                                                      DE100360
                 ERR_NODE= -1;                                         DE100370
                 RETURN;                                               DE100380
              END;                                                     DE100390
      PTLTP-> TWOLEV_TREE.RECORDID= RECORDID;                          DE100400
                                                                       DE100410
                                                                       DE100420
 END DE19;                                                             DE100430
```

```
*PROCESS NAME('MD2'), INCLUDE, F(I);                                    MD200010
 MD2: PROC (UAP, BASIC_OP, MVNIP, MNNIP, OMNIP,                         MD200020
             PTLTP, NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE,            MD200030
             IDARR_CTR, CP, CT_CTR);                                    MD200040
                                                                        MD200050
                                                                        MD200060
 /* MD2= MODIFY2 */                                                     MD200070
                                                                        MD200080
                                                                        MD200090
      %INCLUDE TWOLEV;                                                  MD200100
      %INCLUDE MVNI;                                                    MD200110
      %INCLUDE UA;                                                      MD200120
                                                                        MD200130
                                                                        MD200140
      DCL P PTR;                                                        MD200150
      DCL CLN ENTRY EXTERNAL RETURNS(FIXED),                            MD200160
          CMN ENTRY EXTERNAL,                                           MD200170
          QCFLID ENTRY EXTERNAL RETURNS(FIXED BIN(31)),                 MD200180
          GNI ENTRY EXTERNAL,                                           MD200190
          ILN ENTRY EXTERNAL RETURNS(BIT);                              MD200200
                                                                        MD200210
                                                                        MD200220
      DCL UAP PTR,                                                      MD200230
          BASIC_OP CHAR(1),                                             MD200240
          (MVNIP, MNNIP, OMNIP, PTLTP) PTR,                             MD200250
          (IDARR_CTR, NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE) FIXED,   MD200260
          CP PTR,                                                       MD200270
          CT_CTR FIXED;                                                 MD200280
                                                                        MD200290
                                                                        MD200300
      DCL (NO_LEVEL2NODE, NO_FDUPD, VIA_CTR, ND_CTR) FIXED INIT(0);     MD200310
                                                                        MD200320
                                                                        MD200330
      NO_LEVEL2NODE= CLN(UAP);                                          MD200340
      CALL CMN(UAP, BASIC_OP, NO_MTO1VNODE, NO_MTO1NNODE, CP, CT_CTR);  MD200350
      NO_FDUPD= NO_MTO1VNODE + NO_MTO1NNODE;                            MD200360
      NO_1TOMNODE= NO_LEVEL2NODE - NO_FDUPD;                            MD200370
              DO ND_CTR= 1 TO UAP-> UPDN_ARG.N;                         MD200380
                 IF ILN(UAP, ND_CTR)                                    MD200390
                    & UAP->UPDN_ARG.NODE_DESCRIP(ND_CTR).OP= 'I'        MD200400
                    THEN NO_1TOMNODE= NO_1TOMNODE - 1;                  MD200410
              END;                                                      MD200420
                                                                        MD200430
      IF NO_FDUPD= 0                                                    MD200440
         THEN NO_FDUPD= 1;                                              MD200450
      ALLOCATE TWOLEV_TREE SET (PTLTP);                                 MD200460
      PTLTP-> TWOLEV_TREE.BASIC_OP= ' ';                                MD200470
      PTLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;                              MD200480
      PTLTP-> TWOLEV_TREE.ROOT_FILEID= 0;                               MD200490
      PTLTP-> TWOLEV_TREE.RECORDID= 0;                                  MD200500
      PTLTP-> TWOLEV_TREE.ID_ARR(*).FIELDID= 0;                         MD200510
      PTLTP-> TWOLEV_TREE.ID_ARR(*).DATA= ' ';                          MD200520
                                                                        MD200530
                                                                        MD200540
      PTLTP-> TWOLEV_TREE.BASIC_OP= BASIC_OP;                           MD200550
```

```
      PTLTP-> TWOLEV_TREE.ROOT_FILEID= QCFLID(CP, CT_CTR);                MD200560
      CALL GNI(UAP, BASIC_OP, MVNIP, MNNIP, OMNIP,                        MD200570
               NO_MTO1VNODE, NO_MTO1NNODE, NO_1TOMNODE, CP, CT_CTR);      MD200580
      DO VIA_CTR= 1 TO NO_MTO1VNODE;                                      MD200590
         IDARR_CTR= IDARR_CTR + 1;                                        MD200600
         PTLTP-> TWOLEV_TREE.ID_ARR(IDARR_CTR).FIELDID                    MD200610
            = MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).FIELDID;           MD200620
         PTLTP-> TWOLEV_TREE.ID_ARR(IDARR_CTR).DATA                       MD200630
            = MVNIP-> MTO1VNODE_INFO.INFO_ARR(VIA_CTR).DATA;              MD200640
      END;                                                                MD200650
/*    FREE MVNIP-> MTO1VNODE_INFO  */                                     MD200660
                                                                          MD200670
                                                                          MD200680
END MD2;                                                                  MD200690
```

```
*PROCESS NAME('MD19'), INCLUDE, F(I);                                   MD100010
 MD19: PROC (UAP, MNNIP, PTLTP, NO_MTO1NNODE,                           MD100020
                              IDARR_CTR, ERR_NODE, CP, FP, CT_NO);      MD100030
                                                                        MD100040
 /* MD19= MODIFY19 */                                                   MD100050
 /* INITIAL_NODE_NO SHOULD BE SET TO 0 INSTEAD OF 1                    MD100060
    BUT FOR THE SAKE OF SIMPLICITY DURING RETRIEVAL                     MD100070
    OF RECORDID, IT HAS BEEN SET TO 1.                                  MD100080
    DURING CHECKING OF VALID RECORDID, IT CHECKS                        MD100090
    FILE(ROOT_FILEID).RECORD(2).FIELD(FIELDID OF FIRST NODE) */         MD100100
                                                                        MD100110
 /* THERE ARE 7 CASES TO TEST FOR:                                     MD100120
    1. RECORDID FOR ROOT_NODE NOT IN                                    MD100130
    2. RCDID FOR R_N IN, RCDID FOR MTO1 NODE NOT IN                     MD100140
    3. RCDID FOR R_N IN, RCDID FOR MTO1 NODE IN, OP IS 'R'              MD100150
    4. RCDID FOR R_N IN, RCDID FOR MTO1 NODE IN, OP IS NOT 'R'          MD100160
          RECORD IN                                                     MD100170
    5. RCDID FOR R_N IN, RCDID FOR MTO1 NODE IN, OP IS NOT 'R',         MD100180
          RECORD NOT IN                                                 MD100190
    6. MIXED ORDERED 'R' AND 'A' MTO1NNODES    */                       MD100200
                                                                        MD100210
                                                                        MD100220
                                                                        MD100230
      %INCLUDE MNNI;                                                    MD100240
      %INCLUDE UA;                                                      MD100250
      %INCLUDE TWOLEV;                                                  MD100260
                                                                        MD100270
                                                                        MD100280
      DCL ILN ENTRY EXTERNAL RETURNS(BIT);                              MD100290
      DCL GTRCID ENTRY EXTERNAL RETURNS(FIXED);                         MD100300
                                                                        MD100310
                                                                        MD100320
      DCL P PTR;                                                        MD100330
      DCL (UAP, MNNIP, PTLTP) PTR,                                      MD100340
          (NO_MTO1NNODE, IDARR_CTR, ERR_NODE) FIXED,                    MD100350
          (CP, FP) PTR,                                                 MD100360
          CT_NO FIXED;                                                  MD100370
                                                                        MD100380
                                                                        MD100390
      DCL (INITIAL_NODE_NO, RTNTND_CTR, FINAL_NODE_NO, NIA_CTR,         MD100400
             RC_RTNCODE, RUN) FIXED INIT(0),                            MD100410
          RECORDID FIXED BIN(31) INIT(0);                               MD100420
                                                                        MD100430
                                                                        MD100440
                                                                        MD100450
                                                                        MD100460
                                                                        MD100470
      INITIAL_NODE_NO= 1;                                               MD100480
      DO RTNTND_CTR= 1 TO UAP-> UPDN_ARG.N;                             MD100490
         IF UAP-> UPDN_ARG.NODE_DESCRIP(RTNTND_CTR).OP^= 'I'            MD100500
            THEN DO;                                                    MD100510
                    FINAL_NODE_NO= RTNTND_CTR - 1;                      MD100520
                    RTNTND_CTR= UAP-> UPDN_ARG.N + 1;                   MD100530
                 END;                                                   MD100540
      END;                                                              MD100550
```

```
      RC_RTNCODE= GTRCID(UAP, INITIAL_NODE_NO, FINAL_NODE_NO, RECORDID, MD100560
                         CP, FP, CT_NO);                                MD100570
      IF RC_RTNCODE^= 0                                                 MD100580
         THEN DO;                                                       MD100590
                 ERR_NODE= -1;                                          MD100600
                 GOTO RETN;                                             MD100610
              END;                                                      MD100620
      PTLTP-> TWOLEV_TREE.RECORDID= RECORDID;                           MD100630
                                                                        MD100640
                                                                        MD100650
      IF NO_MTO1NNODE^= 0                                               MD100660
      THEN                                                              MD100670
      DO;                                                               MD100680
      DO NIA_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;             MD100690
         INITIAL_NODE_NO= MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).     MD100700
                          INITIAL_NODE_NO;                              MD100710
         DO RTNTND_CTR= INITIAL_NODE_NO + 1 TO UAP-> UPDN_ARG.N;        MD100720
           IF RTNTND_CTR= UAP-> UPDN_ARG.N                              MD100730
              THEN MNNIP->MTO1NNODE_INFO.INFO_ARR(NIA_CTR).             MD100740
                      FINAL_NODE_NO= RTNTND_CTR;                        MD100750
       ELSE IF ILN(UAP, RTNTND_CTR)                                     MD100760
              THEN DO;                                                  MD100770
                      MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).         MD100780
                         FINAL_NODE_NO= RTNTND_CTR - 1;                 MD100790
                      RTNTND_CTR= UAP-> UPDN_ARG.N + 1;                 MD100800
                   END;                                                 MD100810
         END;                                                           MD100820
      END;                                                              MD100830
      RUN= 1;                                                           MD100840
LOOP:                                                                   MD100850
      DO NIA_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;             MD100860
         INITIAL_NODE_NO= MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).     MD100870
                          INITIAL_NODE_NO;                              MD100880
         IF RUN= 1                                                      MD100890
            THEN IF UAP-> UPDN_ARG.NODE_DESCRIP(INITIAL_NODE_NO).       MD100900
                         OP= 'A'                                        MD100910
                 THEN GOTO EOL;                                         MD100920
         IF RUN= 2                                                      MD100930
            THEN IF UAP-> UPDN_ARG.NODE_DESCRIP(INITIAL_NODE_NO).       MD100940
                         OP= 'R'                                        MD100950
                 THEN GOTO EOL;                                         MD100960
         IF UAP-> UPDN_ARG.NODE_DESCRIP(INITIAL_NODE_NO).OP= 'R'        MD100970
            THEN RECORDID=0;                                            MD100980
            ELSE DO;                                                    MD100990
                    FINAL_NODE_NO= MNNIP->MTO1NNODE_INFO.INFO_ARR       MD101000
                                   (NIA_CTR).FINAL_NODE_NO;             MD101010
                    RC_RTNCODE= GTRCID(UAP, INITIAL_NODE_NO,            MD101020
                                       FINAL_NODE_NO, RECORDID,         MD101030
                                       CP, FP, CT_NO);                  MD101040
                    IF RC_RTNCODE^= 0                                   MD101050
                       THEN DO;                                         MD101060
                               ERR_NODE= INITIAL_NODE_NO;               MD101070
                               GOTO RETN;                               MD101080
                            END;                                        MD101090
                 END;                                                   MD101100
```

```
         IDARR_CTR= IDARR_CTR + 1;                                  MD101110
         PTLTP-> TWOLEV_TREE.ID_ARR(IDARR_CTR).FIELDID              MD101120
            = MNNIP-> MTO1NNODE_INFO.INFO_ARR(NIA_CTR).             MD101130
                FIELDID;                                            MD101140
         PTLTP-> TWOLEV_TREE.ID_ARR(IDARR_CTR).DATA                 MD101150
            = RECORDID;                                             MD101160
EOL:                                                                MD101170
      END;                                                          MD101180
      IF RUN= 1                                                     MD101190
         THEN DO;                                                   MD101200
                 RUN= 2;                                            MD101210
                 GOTO LOOP;                                         MD101220
              END;                                                  MD101230
      END;                                                          MD101240
RETN:                                                               MD101250
/*    FREE MNNIP-> MTO1NNODE_INFO */                                MD101260
                                                                    MD101270
                                                                    MD101280
END MD19;                                                           MD101290
```

```
*PROCESS NAME('MD20'), INCLUDE, F(I);                                   MD200010
 /* MD20= MODIFY20 */                                                  MD200020
 /* 2 CASES TO TEST FOR: VALID AND INVALID RECORDID */                 MD200030
 /* ALSO TEST FOR PROPER REORDERING OF 'R' AND 'A' NODES               MD200040
    THE CHANGES ARE SPCT, SFLAPC TO SFLAPC1, AND ZGTCT */             MD200050
                                                                       MD200060
                                                                       MD200070
 MD20: PROC (UAP, OMNIP, PTLTP, ERR_NODE, PAP, CP, FP, CT_NO);         MD200080
                                                                       MD200090
                                                                       MD200100
      %INCLUDE UA;                                                     MD200110
      %INCLUDE TWOLEV;                                                 MD200120
      %INCLUDE ONI;                                                    MD200130
      %INCLUDE PA;                                                     MD200140
                                                                       MD200150
                                                                       MD200160
      DCL NULL BUILTIN;                                                MD200170
      DCL QCFLDS ENTRY EXTERNAL RETURNS(FIXED BIN(31));                MD200180
      DCL ILN ENTRY EXTERNAL RETURNS(BIT);                             MD200190
      DCL GTRCID ENTRY EXTERNAL RETURNS(FIXED);                        MD200200
      DCL QCFDID ENTRY EXTERNAL RETURNS(FIXED BIN(31));                MD200210
                                                                       MD200220
                                                                       MD200230
      DCL P PTR;                                                       MD200240
      DCL (UAP, OMNIP, PTLTP) PTR,                                     MD200250
          ERR_NODE FIXED,                                              MD200260
          (PAP, CP, FP) PTR,                                           MD200270
          CT_NO FIXED;                                                 MD200280
                                                                       MD200290
                                                                       MD200300
      DCL BASIC_OP CHAR(1) INIT(' '),                                  MD200310
          NO_FDUPD FIXED INIT(0),                                      MD200320
          ROOT_RECORDID FIXED BIN(31) INIT(0),                         MD200330
          RUN FIXED INIT(0),                                           MD200340
          (ONIA_CTR, NO_1TOMNODE, NODE_NO) FIXED INIT(0),              MD200350
          BSETID FIXED BIN(15) INIT(0),                                MD200360
          STLTP PTR INIT(NULL()),                                      MD200370
          (INITIAL_NODE_NO, RTNTND_CTR, FINAL_NODE_NO, RC_RTNCODE)     MD200380
             FIXED INIT(0),                                            MD200390
          RECORDID FIXED BIN(31) INIT(0),                              MD200400
          PA_CTR FIXED INIT(0);                                        MD200410
                                                                       MD200420
                                                                       MD200430
      BASIC_OP= 'M';                                                   MD200440
      NO_FDUPD= 1;                                                     MD200450
      ROOT_RECORDID= PTLTP-> TWOLEV_TREE.RECORDID;                     MD200460
      RUN=1;                                                           MD200470
 LOOP:                                                                 MD200480
      DO ONIA_CTR= 1 TO OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;             MD200490
         NODE_NO= OMNIP-> OTOMNODE_INFO.INFO_ARR(ONIA_CTR);            MD200500
         IF RUN=1                                                      MD200510
            THEN IF UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).OP= 'A'       MD200520
                    THEN GOTO EOL;                                     MD200530
         IF RUN=2                                                      MD200540
            THEN IF UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).OP= 'R'       MD200550
```

```
                THEN GOTO EOL;                                          MD200560
        BSETID= UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).BSETID;            MD200570
        ALLOCATE TWOLEV_TREE SET (STLTP);                               MD200580
        STLTP-> TWOLEV_TREE.BASIC_OP= ' ';                              MD200590
        STLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;                            MD200600
        STLTP-> TWOLEV_TREE.ROOT_FILEID= 0;                             MD200610
        STLTP-> TWOLEV_TREE.RECORDID= 0;                                MD200620
        STLTP-> TWOLEV_TREE.ID_ARR.FIELDID= 0;                          MD200630
        STLTP-> TWOLEV_TREE.ID_ARR.DATA= ' ';                           MD200640
        STLTP-> TWOLEV_TREE.BASIC_OP= BASIC_OP;                         MD200650
        STLTP-> TWOLEV_TREE.ROOT_FILEID= QCFLDS(CP, CT_NO, BSETID);     MD200660
        INITIAL_NODE_NO= NODE_NO;                                       MD200670
        DO RTNTND_CTR= INITIAL_NODE_NO + 1 TO UAP-> UPDN_ARG.N;         MD200680
           IF RTNTND_CTR= UAP-> UPDN_ARG.N                              MD200690
              THEN FINAL_NODE_NO= RTNTND_CTR;                           MD200700
      ELSE IF ILN(UAP, RTNTND_CTR)                                      MD200710
              THEN DO;                                                  MD200720
                      FINAL_NODE_NO= RTNTND_CTR - 1;                    MD200730
                      RTNTND_CTR= UAP-> UPDN_ARG.N + 1;                 MD200740
                   END;                                                 MD200750
        END;                                                            MD200760
        RC_RTNCODE= GTRCID(UAP, INITIAL_NODE_NO, FINAL_NODE_NO,         MD200770
                           RECORDID, CP, FP, CT_NO);                    MD200780
        IF RC_RTNCODE^= 0                                               MD200790
           THEN DO;                                                     MD200800
                   ERR_NODE= INITIAL_NODE_NO;                           MD200810
                   GOTO RETN;                                           MD200820
                END;                                                    MD200830
        STLTP-> TWOLEV_TREE.RECORDID= RECORDID;                         MD200840
        STLTP-> TWOLEV_TREE.ID_ARR(1).FIELDID= QCFDID(CP, CT_NO,        MD200850
                                                      BSETID);          MD200860
        STLTP-> TWOLEV_TREE.ID_ARR(1).DATA= ROOT_RECORDID;              MD200870
        PA_CTR= PA_CTR + 1;                                             MD200880
        PAP-> PTR_ARR.ARR(PA_CTR)= STLTP;                               MD200890
EOL:                                                                    MD200900
     END;                                                               MD200910
     IF RUN= 1                                                          MD200920
        THEN DO;                                                        MD200930
                RUN= 2;                                                 MD200940
                GOTO LOOP;                                              MD200950
             END;                                                       MD200960
RETN:                                                                   MD200970
/*    FREE OMNIP-> OTOMNODE_INFO */                                     MD200980
                                                                        MD200990
                                                                        MD201000
                                                                        MD201010
                                                                        MD201020
END MD20;                                                               MD201030
```

```
*PROCESS NAME('ZGTCT'), INCLUDE, F(I);                              ZGT00010
 ZGTCT: PROC(CP, ROOT_PSETID) RETURNS(FIXED);                       ZGT00020
 /* ZGTCT= GET_CAT_ENTRY */                                         ZGT00030
                                                                    ZGT00040
                                                                    ZGT00050
      %INCLUDE SPCT;                                                ZGT00060
      DCL P PTR;                                                    ZGT00070
      DCL CP PTR,                                                   ZGT00080
          ROOT_PSETID FIXED BIN(15);                                ZGT00090
      DCL CT_NO FIXED INIT(0);                                      ZGT00100
                                                                    ZGT00110
                                                                    ZGT00120
      DO CT_NO= 1 TO 5;                                             ZGT00130
         IF CP->PCAT(CT_NO).PSETID= ROOT_PSETID                     ZGT00140
            THEN RETURN(CT_NO);                                     ZGT00150
      END;                                                          ZGT00160
                                                                    ZGT00170
                                                                    ZGT00180
 END ZGTCT;                                                         ZGT00190
```

```
*PROCESS NAME('TPROC'), INCLUDE, F(I);                                  TPR00010
 TPROC: PROC(PTLTP);                                                    TPR00020
 /* TPROC= THE INTER-LEVEL CALLING PROCEDURE */                         TPR00030
                                                                        TPR00040
                                                                        TPR00050
      %INCLUDE TWOLEV;                                                  TPR00060
                                                                        TPR00070
                                                                        TPR00080
      DCL MOD BUILTIN;                                                  TPR00090
      DCL P PTR;                                                        TPR00100
      DCL PTLTP PTR;                                                    TPR00110
      DCL RANDOM1 FIXED INIT(0),                                        TPR00120
          RANDOM2 FIXED INIT(0),                                        TPR00130
          RANDOM3 FIXED INIT(0),                                        TPR00140
          RANDOM FIXED INIT(0);                                         TPR00150
                                                                        TPR00160
      IF PTLTP-> TWOLEV_TREE.RECORDID= 0                                TPR00170
            & PTLTP-> TWOLEV_TREE.BASIC_OP= 'C'                         TPR00180
         THEN PTLTP-> TWOLEV_TREE.RECORDID= 4;                          TPR00190
      SELECT(PTLTP-> TWOLEV_TREE.BASIC_OP);                             TPR00200
         WHEN('C') RANDOM1= 0;                                          TPR00210
         WHEN('D') RANDOM1= 1;                                          TPR00220
         WHEN('M') RANDOM1= 2;                                          TPR00230
      END;                                                              TPR00240
                                                                        TPR00250
         RANDOM2= MOD(PTLTP-> TWOLEV_TREE.ROOT_FILEID, 3);              TPR00260
         RANDOM3= PTLTP-> TWOLEV_TREE.NO_OF_FIELD;                      TPR00270
         RANDOM= MOD(RANDOM1 + RANDOM2 + RANDOM3, 5);                   TPR00280
      SELECT(RANDOM);                                                   TPR00290
         WHEN(0, 1, 2, 3) PTLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;          TPR00300
         WHEN(4) PTLTP-> TWOLEV_TREE.B_OP_RTNCODE= 0;                   TPR00310
      END;                                                              TPR00320
                                                                        TPR00330
 END TPROC;                                                             TPR00340
```

```
*PROCESS NAME('CONTROL'), INCLUDE, F(I);                                CON00010
 CONTROL: PROC(UAP) RETURNS(PTR);                                       CON00020
                                                                        CON00030
                                                                        CON00040
      %INCLUDE UA;                                                      CON00050
 /* UP TO CMN */                                                       CON00060
                                                                        CON00070
      %INCLUDE TWOLEV;                                                  CON00080
      %INCLUDE MVNI;                                                    CON00090
      %INCLUDE MNNI;                                                    CON00100
      %INCLUDE ONI;                                                     CON00110
 /* UP TO CRMD2 AND GNI */                                             CON00120
                                                                        CON00130
                                                                        CON00140
      %INCLUDE PA;                                                      CON00150
 /* UP TO CR20 */                                                      CON00160
                                                                        CON00170
                                                                        CON00180
      %INCLUDE UR;                                                      CON00190
 /* FINISHING TOUCH */                                                 CON00200
                                                                        CON00210
                                                                        CON00220
      DCL P PTR;                                                        CON00230
      DCL SBDUA ENTRY EXTERNAL RETURNS(PTR);                            CON00240
      DCL CLN ENTRY EXTERNAL RETURNS(FIXED);                            CON00250
      DCL NULL BUILTIN;                                                 CON00260
      DCL UAP PTR,                                                      CON00270
          NO_LEVEL2NODE FIXED INIT(0);                                  CON00280
 /* UP TO CLN */                                                       CON00290
                                                                        CON00300
      DCL SFLAPC ENTRY EXTERNAL,                                        CON00310
          ZGTCT ENTRY EXTERNAL RETURNS(FIXED),                          CON00320
          CMN ENTRY EXTERNAL;                                           CON00330
      DCL (FP, CP) PTR INIT(NULL()),                                    CON00340
          CT_NO FIXED INIT(0),                                          CON00350
          (NO_MTO1VNODE, NO_MTO1NNODE) FIXED INIT(0);                   CON00360
 /* UP TO CMN */                                                       CON00370
                                                                        CON00380
      DCL (CR2, GNI) ENTRY EXTERNAL;                                    CON00390
      DCL (MVNIP, MNNIP, OMNIP, PTLTP) PTR INIT(NULL()),                CON00400
          (IDARR_CTR, NO_1TOMNODE, PR_CTR) FIXED INIT(0);               CON00410
 /* UP TO CR2 AND GNI */                                               CON00420
                                                                        CON00430
                                                                        CON00440
      DCL DE2 ENTRY EXTERNAL;                                           CON00450
 /* UP TO DE2 */                                                       CON00460
                                                                        CON00470
                                                                        CON00480
      DCL MD2 ENTRY EXTERNAL;                                           CON00490
 /* UP TO MD2 */                                                       CON00500
                                                                        CON00510
                                                                        CON00520
      DCL CR19 ENTRY EXTERNAL;                                          CON00530
      DCL ERR_NODE FIXED INIT(0);                                       CON00540
 /* UP TO CR19 */                                                      CON00550
```

```
                                                                    CON00560
                                                                    CON00570
     DCL DE19 ENTRY EXTERNAL;                                       CON00580
/* UP TO DE19 */                                                   CON00590
                                                                    CON00600
                                                                    CON00610
     DCL MD19 ENTRY EXTERNAL;                                       CON00620
/* UP TO MD19 */                                                   CON00630
                                                                    CON00640
                                                                    CON00650
     DCL CR20 ENTRY EXTERNAL;                                       CON00660
     DCL PA_CTR FIXED INIT(0),                                      CON00670
         PAP PTR INIT(NULL());                                      CON00680
/* UP TO CR20 */                                                   CON00690
                                                                    CON00700
                                                                    CON00710
     DCL MD20 ENTRY EXTERNAL;                                       CON00720
/* UP TO MD20 */                                                   CON00730
                                                                    CON00740
     DCL TPROC ENTRY EXTERNAL;                                      CON00750
     DCL URP PTR INIT(NULL());                                      CON00760
/* FINISHING TOUCH */                                              CON00770
                                                                    CON00780
                                                                    CON00790
     UAP= SBDUA();                                                  CON00800
     NO_LEVEL2NODE= CLN(UAP);                                       CON00810
     PUT SKIP LIST('NUMBER OF LEVEL 2 NODES IS', NO_LEVEL2NODE);    CON00820
/* UP TO CLN */                                                    CON00830
                                                                    CON00840
                                                                    CON00850
     CALL SFLAPC(FP, CP);                                           CON00860
     CT_NO= ZGTCT(CP, UAP-> UPDN_ARG.ROOT_PSETID);                  CON00870
     CALL CMN(UAP, UAP-> UPDN_ARG.BASIC_OP, NO_MTO1VNODE, NO_MTO1NNODE,CON00880
              CP, CT_NO);                                           CON00890
     PUT SKIP LIST('NUMBER OF MTO1 VALUE NODES IS', NO_MTO1VNODE);  CON00900
     PUT SKIP LIST('NUMBER OF MTO1 ENTITY NODES IS', NO_MTO1NNODE); CON00910
/* UP TO CMN */                                                    CON00920
    SELECT(UAP-> UPDN_ARG.BASIC_OP);                                CON00930
       WHEN ('C')                                                   CON00940
           CALL CR2(UAP, UAP-> UPDN_ARG.BASIC_OP, MVNIP, MNNIP,     CON00950
                    OMNIP, PTLTP, NO_MTO1VNODE, NO_MTO1NNODE,       CON00960
                    NO_1TOMNODE, CP, CT_NO);                        CON00970
       WHEN ('D')                                                   CON00980
           CALL DE2(UAP-> UPDN_ARG.BASIC_OP, PTLTP, CP, CT_NO);     CON00990
       WHEN ('M')                                                   CON01000
           CALL MD2(UAP, UAP-> UPDN_ARG.BASIC_OP, MVNIP, MNNIP,     CON01010
                    OMNIP, PTLTP, NO_MTO1VNODE, NO_MTO1NNODE,       CON01020
                    NO_1TOMNODE, IDARR_CTR, CP, CT_NO);             CON01030
    END;                                                            CON01040
                                                                    CON01050
                                                                    CON01060
                                                                    CON01070
                                                                    CON01080
     SELECT(UAP-> UPDN_ARG.BASIC_OP);                               CON01090
     WHEN ('C', 'M')                                                CON01100
```

```
      DO;                                                                  CONO1110
      PUT SKIP LIST('PRINTING CONTENTS OF TWOLEV_TREE');                   CONO1120
      PUT SKIP LIST('TLT.BASIC_OP', PTLTP-> TWOLEV_TREE.BASIC_OP);         CONO1130
      PUT SKIP LIST('TLT.ROOT_FILEID', PTLTP-> TWOLEV_TREE.ROOT_FILEID);CONO1140
      DO PR_CTR= 1 TO PTLTP-> TWOLEV_TREE.NO_OF_FIELD;                     CONO1150
         PUT SKIP LIST('ENTRY', PR_CTR, ':', 'FIELDID AND DATA');          CONO1160
         PUT SKIP LIST(PTLTP-> TWOLEV_TREE.ID_ARR(PR_CTR).FIELDID,         CONO1170
                       PTLTP-> TWOLEV_TREE.ID_ARR(PR_CTR).DATA);           CONO1180
      END;                                                                 CONO1190
      IF NO_MTO1VNODE^= 0                                                  CONO1200
      THEN                                                                 CONO1210
      DO;                                                                  CONO1220
      PUT SKIP LIST('PRINTING CONTENTS OF MVNI');                          CONO1230
      DO PR_CTR= 1 TO MVNIP-> MTO1VNODE_INFO.NO_MTO1VNODE;                 CONO1240
         PUT SKIP LIST('ENTRY', PR_CTR, ':', 'NODE_NO, FIELDID, DATA');    CONO1250
         PUT SKIP LIST(MVNIP-> MTO1VNODE_INFO.INFO_ARR(PR_CTR).NODE_NO,    CONO1260
                       MVNIP-> MTO1VNODE_INFO.INFO_ARR(PR_CTR).FIELDID,    CONO1270
                       MVNIP-> MTO1VNODE_INFO.INFO_ARR(PR_CTR).DATA);      CONO1280
      END;                                                                 CONO1290
      END;                                                                 CONO1300
      IF NO_MTO1NNODE^= 0                                                  CONO1310
      THEN                                                                 CONO1320
      DO;                                                                  CONO1330
      PUT SKIP LIST('PRINTING CONTENTS OF MNNI');                          CONO1340
      DO PR_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;                 CONO1350
         PUT SKIP LIST('ENTRY', PR_CTR,':', 'INITIAL_NODE_NO, FIELDID');CONO1360
         PUT SKIP LIST(MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).            CONO1370
                          INITIAL_NODE_NO,                                 CONO1380
                       MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).FIELDID);CONO1390
      END;                                                                 CONO1400
      END;                                                                 CONO1410
      IF NO_1TOMNODE^= 0                                                   CONO1420
      THEN                                                                 CONO1430
      DO;                                                                  CONO1440
      PUT SKIP LIST('PRINTING CONTENTS OF ONI');                           CONO1450
      DO PR_CTR= 1 TO OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;                   CONO1460
         PUT SKIP LIST('ENTRY', PR_CTR, ':', 'NODE_NO');                   CONO1470
         PUT SKIP LIST(OMNIP-> OTOMNODE_INFO.INFO_ARR(PR_CTR));            CONO1480
      END;                                                                 CONO1490
      END;                                                                 CONO1500
/* UP TO CRMD2 AND GNI */                                                  CONO1510
      END;                                                                 CONO1520
                                                                           CONO1530
                                                                           CONO1540
      WHEN ('D')                                                           CONO1550
      DO;                                                                  CONO1560
      PUT SKIP LIST('PRINTING CONTENTS OF TWOLEV_TREE');                   CONO1570
      PUT SKIP LIST('TLT.BASIC_OP', PTLTP->TWOLEV_TREE.BASIC_OP);          CONO1580
      PUT SKIP LIST('TLT.ROOT_FILEID', PTLTP->TWOLEV_TREE.ROOT_FILEID);  CONO1590
/* UP TO DE2 */                                                          CONO1600
      END;                                                                 CONO1610
      END;                                                                 CONO1620
                                                                           CONO1630
      SELECT(UAP-> UPDN_ARG.BASIC_OP);                                     CONO1640
         WHEN('C')                                                         CONO1650
```

```
         IF NO_MTO1NNODE^= 0                                          CONO1660
         THEN                                                         CONO1670
         CALL CR19(UAP, MNNIP, PTLTP, ERR_NODE,                       CONO1680
                   CP, FP, CT_NO);                                    CONO1690
      WHEN('D')                                                       CONO1700
         CALL DE19(UAP, PTLTP, ERR_NODE, CP, FP, CT_NO);              CONO1710
      WHEN('M')                                                       CONO1720
         CALL MD19(UAP, MNNIP, PTLTP, NO_MTO1NNODE, IDARR_CTR,        CONO1730
                   ERR_NODE, CP, FP, CT_NO);                          CONO1740
   END;                                                               CONO1750
                                                                      CONO1760
                                                                      CONO1770
   SELECT(UAP-> UPDN_ARG.BASIC_OP);                                   CONO1780
   WHEN ('C')                                                         CONO1790
   DO;                                                                CONO1800
   IF ERR_NODE= 0                                                     CONO1810
   THEN                                                               CONO1820
   DO;                                                                CONO1830
   PUT SKIP LIST('PRINTING CONTENTS OF TWOLEV_TREE');                 CONO1840
   DO PR_CTR= 1 TO PTLTP-> TWOLEV_TREE.NO_OF_FIELD;                   CONO1850
      PUT SKIP LIST('ENTRY', PR_CTR, ':', 'FIELDID AND DATA');        CONO1860
      PUT SKIP LIST(PTLTP-> TWOLEV_TREE.ID_ARR(PR_CTR).FIELDID,       CONO1870
                    PTLTP-> TWOLEV_TREE.ID_ARR(PR_CTR).DATA);         CONO1880
   END;                                                               CONO1890
   IF NO_MTO1NNODE^= 0                                                CONO1900
   THEN                                                               CONO1910
   DO;                                                                CONO1920
   PUT SKIP LIST('PRINTING CONTENTS OF MNNI');                        CONO1930
   DO PR_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;               CONO1940
      PUT SKIP LIST('ENTRY', PR_CTR,':',                              CONO1950
                    'INITIAL_NODE_NO, FINAL_NODE_NO, FIELDID');       CONO1960
      PUT SKIP LIST(MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).          CONO1970
                    INITIAL_NODE_NO,                                  CONO1980
                 MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).             CONO1990
                    FINAL_NODE_NO,                                    CONO2000
                 MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).FIELDID);    CONO2010
   END;                                                               CONO2020
   END;                                                               CONO2030
   END;                                                               CONO2040
   END;                                                               CONO2050
/* UP TO CR19 */                                                     CONO2060
                                                                      CONO2070
                                                                      CONO2080
   WHEN ('D')                                                         CONO2090
   DO;                                                                CONO2100
   IF ERR_NODE= 0                                                     CONO2110
   THEN                                                               CONO2120
   DO;                                                                CONO2130
   PUT SKIP LIST('PRINTING CONTENTS OF TWOLEV_TREE');                 CONO2140
   PUT SKIP LIST('RECORDID OF ROOT_NODE');                            CONO2150
   PUT SKIP LIST(PTLTP-> TWOLEV_TREE.RECORDID);                       CONO2160
   END;                                                               CONO2170
   END;                                                               CONO2180
/* UP TO DE19 */                                                     CONO2190
                                                                      CONO2200
```

```
                                                                      CONO2210
      WHEN('M')                                                       CONO2220
      DO;                                                             CONO2230
      IF ERR_NODE= 0                                                  CONO2240
      THEN                                                            CONO2250
      DO;                                                             CONO2260
      PUT SKIP LIST('PRINTING CONTENTS OF TWOLEV_TREE');              CONO2270
      PUT SKIP LIST('RECORDID OF ROOT_NODE');                         CONO2280
      PUT SKIP LIST(PTLTP-> TWOLEV_TREE.RECORDID);                    CONO2290
      DO PR_CTR= 1 TO PTLTP-> TWOLEV_TREE.NO_OF_FIELD;                CONO2300
         PUT SKIP LIST('ENTRY', PR_CTR, ':', 'FIELDID AND DATA');     CONO2310
         PUT SKIP LIST(PTLTP-> TWOLEV_TREE.ID_ARR(PR_CTR).FIELDID,    CONO2320
                       PTLTP-> TWOLEV_TREE.ID_ARR(PR_CTR).DATA);      CONO2330
      END;                                                            CONO2340
      IF NO_MTO1NNODE^= 0                                             CONO2350
      THEN                                                            CONO2360
      DO;                                                             CONO2370
      PUT SKIP LIST('PRINTING CONTENTS OF MNNI');                     CONO2380
      DO PR_CTR= 1 TO MNNIP-> MTO1NNODE_INFO.NO_MTO1NNODE;            CONO2390
         PUT SKIP LIST('ENTRY', PR_CTR, ':',                          CONO2400
                       'INITIAL_NODE_NO, FINAL_NODE_NO, FIELDID');    CONO2410
         PUT SKIP LIST(MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).       CONO2420
                       INITIAL_NODE_NO,                               CONO2430
                     MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).         CONO2440
                       FINAL_NODE_NO,                                 CONO2450
                     MNNIP-> MTO1NNODE_INFO.INFO_ARR(PR_CTR).         CONO2460
                       FIELDID);                                      CONO2470
      END;                                                            CONO2480
      END;                                                            CONO2490
      END;                                                            CONO2500
      END;                                                            CONO2510
                                                                      CONO2520
                                                                      CONO2530
      END;                                                            CONO2540
/* UP TO MD19 */                                                      CONO2550
                                                                      CONO2560
                                                                      CONO2570
      ALLOCATE UPDN_RTN SET(URP);                                     CONO2580
      URP-> UPDN_RTN.RTN_CODE= 0;                                     CONO2590
      URP-> UPDN_RTN.ERROR_NODENO= 0;                                 CONO2600
                                                                      CONO2610
      IF ERR_NODE^= 0                                                 CONO2620
         THEN DO;                                                     CONO2630
                URP-> UPDN_RTN.ERROR_NODENO= ERR_NODE;                CONO2640
                PUT SKIP LIST('NODE THAT CAUSED BREAKDOWN IN P-TREE', CONO2650
                              URP-> UPDN_RTN.ERROR_NODENO);           CONO2660
                RETURN(URP);                                          CONO2670
              END;                                                    CONO2680
                                                                      CONO2690
                                                                      CONO2700
                                                                      CONO2710
                                                                      CONO2720
                                                                      CONO2730
      SELECT(UAP-> UPDN_ARG.BASIC_OP);                                CONO2740
      WHEN ('C', 'M')                                                 CONO2750
```

```
      IF NO_1TOMNODE^= 0                                                    CONO2760
      THEN                                                                  CONO2770
      DO;                                                                   CONO2780
/*    PTLTP-> TWOLEV_TREE.RECORDID= 4*//* TO BE MODIFIED IN CONTROL */     CONO2790
      PA_CTR= OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;                            CONO2800
      ALLOCATE PTR_ARR SET(PAP);                                            CONO2810
      PAP-> PTR_ARR.ARR(*)= NULL();                                         CONO2820
      SELECT (UAP-> UPDN_ARG.BASIC_OP);                                     CONO2830
      WHEN ('C')                                                            CONO2840
      CALL CR2O(UAP, OMNIP, PTLTP, ERR_NODE, PAP, CP, FP, CT_NO);           CONO2850
      WHEN ('M')                                                            CONO2860
      CALL MD2O(UAP, OMNIP, PTLTP, ERR_NODE, PAP, CP, FP, CT_NO);           CONO2870
      END;                                                                  CONO2880
      END;                                                                  CONO2890
      WHEN ('D')                                                            CONO2900
      GOTO SUCCESSFUL_UPDATE;                                               CONO2910
      END;                                                                  CONO2920
                                                                            CONO2930
      SELECT(UAP-> UPDN_ARG.BASIC_OP);                                      CONO2940
         WHEN('C', 'M')                                                     CONO2950
          DO;                                                               CONO2960
      IF ERR_NODE= 0                                                        CONO2970
         THEN                                                               CONO2980
          IF NO_1TOMNODE^= 0                                                CONO2990
          THEN                                                              CONO3000
          DO;                                                               CONO3010
      PUT SKIP LIST('PRINTING CONTENTS OF PTR_ARR');                        CONO3020
      DO PR_CTR= 1 TO OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;                    CONO3030
         PUT SKIP LIST('ENTRY', PR_CTR, ':', 'BASIC_OP, FILEID');           CONO3040
         PUT SKIP LIST(PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.BASIC_OP,CONO3050
                       PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.           CONO3060
                             ROOT_FILEID);                                CONO3070
         PUT SKIP LIST('RECORDID, NO_FDUPD');                               CONO3080
         PUT SKIP LIST(PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.             CONO3090
                             RECORDID,                                      CONO3100
                       PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.           CONO3110
                             NO_OF_FIELD);                                CONO3120
         PUT SKIP LIST('FIELDID, DATA');                                    CONO3130
         PUT SKIP LIST(PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.             CONO3140
                             ID_ARR(1).FIELDID,                             CONO3150
                       PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.           CONO3160
                             ID_ARR(1).DATA);                             CONO3170
      END;                                                                  CONO3180
          END;                                                              CONO3190
          END;                                                              CONO3200
      END;                                                                  CONO3210
                                                                            CONO3220
      IF ERR_NODE^= 0                                                       CONO3230
         THEN DO;                                                           CONO3240
                 URP-> UPDN_RTN.ERROR_NODENO= ERR_NODE;                     CONO3250
                 PUT SKIP LIST('NODE THAT CAUSED BREAKDOWN IN S-TREE',      CONO3260
                               URP-> UPDN_RTN.ERROR_NODENO);                CONO3270
                 RETURN(URP);                                               CONO3280
              END;                                                          CONO3290
      IF PTLTP-> TWOLEV_TREE.NO_OF_FIELD= 1                                 CONO3300
```

```
        THEN IF PTLTP-> TWOLEV_TREE.BASIC_OP= 'M'                    CONO3310
                & PTLTP-> TWOLEV_TREE.ID_ARR(1).FIELDID= 0           CONO3320
                THEN DO;                                             CONO3330
                     PUT SKIP LIST('NO PRIMARY TREE PROCESSED');     CONO3340
                     GOTO PROCESS_1TOM;                              CONO3350
                     END;                                            CONO3360
                ELSE DO;                                             CONO3370
                     CALL TPROC(PTLTP);                              CONO3380
                     PUT SKIP LIST('PRIMARY TREE PROCESSED');        CONO3390
                     END;                                            CONO3400
        ELSE DO;                                                     CONO3410
             CALL TPROC(PTLTP);                                      CONO3420
             PUT SKIP LIST('PRIMARY TREE PROCESSED');                CONO3430
             END;                                                    CONO3440
PROCESS_1TOM:                                                        CONO3450
     IF PTLTP-> TWOLEV_TREE.B_OP_RTNCODE^= 0                         CONO3460
        THEN DO;                                                     CONO3470
                URP-> UPDN_RTN.RTN_CODE= PTLTP-> TWOLEV_TREE.        CONO3480
                                                 B_OP_RTNCODE;       CONO3490
                PUT SKIP LIST('OPERATION FOR MTO1 V AND N NODES');   CONO3500
                PUT SKIP LIST('ABORRTED, RETURN CODE IS',            CONO3510
                              URP-> UPDN_RTN.RTN_CODE);              CONO3520
                RETURN(URP);                                         CONO3530
             END;                                                    CONO3540
                                                                     CONO3550
     PUT SKIP LIST('OPERATION FOR MTO1 V AND N NODES SUCCESSFUL');   CONO3560
     PUT SKIP LIST('BASIC_OP RTNCODE AND ERROR_NODENO');             CONO3570
     PUT SKIP LIST(URP->UPDN_RTN.RTN_CODE, URP->UPDN_RTN.ERROR_NODENO);CONO3580
                                                                     CONO3590
     IF NO_1TOMNODE^= 0                                              CONO3600
     THEN                                                            CONO3610
     DO;                                                             CONO3620
     IF UAP-> UPDN_ARG.BASIC_OP= 'C'                                 CONO3630
        THEN DO PR_CTR= 1 TO OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;      CONO3640
                PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.ID_ARR(1).DATA CONO3650
                   = PTLTP-> TWOLEV_TREE.RECORDID;                   CONO3660
                PUT SKIP LIST('NEW RECORD CREATED:');                CONO3670
                PUT SKIP LIST('ENTRY', PR_CTR, ':', 'FIELDID, DATA');CONO3680
                PUT SKIP LIST(PAP-> PTR_ARR.ARR(PR_CTR)->            CONO3690
                              TWOLEV_TREE.ID_ARR(1).FIELDID,         CONO3700
                              PAP-> PTR_ARR.ARR(PR_CTR)->            CONO3710
                              TWOLEV_TREE.ID_ARR(1).DATA);           CONO3720
             END;                                                    CONO3730
     DO PR_CTR= 1 TO OMNIP-> OTOMNODE_INFO.NO_1TOMNODE;              CONO3740
        CALL TPROC(PAP-> PTR_ARR.ARR(PR_CTR));                       CONO3750
        IF PAP-> PTR_ARR.ARR(PR_CTR)-> TWOLEV_TREE.B_OP_RTNCODE      CONO3760
             ^= 0                                                    CONO3770
           THEN DO;                                                  CONO3780
                   URP-> UPDN_RTN.RTN_CODE                         CONO3790
                      = PAP-> PTR_ARR.ARR(PR_CTR)->                  CONO3800
                           TWOLEV_TREE.B_OP_RTNCODE;                 CONO3810
                   PUT SKIP LIST('OPERATION NUMBER', PR_CTR,       CONO3820
                                 'ABORTED, RETURN CODE IS',        CONO3830
                                 URP-> UPDN_RTN.RTN_CODE);         CONO3840
                   RETURN(URP);                                      CONO3850
```

```
                END;                                                    CONO3860
                                                                        CONO3870
                                                                        CONO3880
                                                                        CONO3890
     END;                                                               CONO3900
     END;                                                               CONO3910
                                                                        CONO3920
SUCCESSFUL_UPDATE:                                                      CONO3930
     PUT SKIP LIST('OPERATION COMPLETELY SUCCESSFUL');                  CONO3940
     PUT SKIP LIST('BASIC_OP RETNCODE, ERROR_NODENO');                  CONO3950
     PUT SKIP LIST(URP-> UPDN_RTN.RTN_CODE,                             CONO3960
                   URP-> UPDN_RTN.ERROR_NODENO);                        CONO3970
     RETURN(URP);                                                       CONO3980
END CONTROL;                                                            CONO3990
```

```
*PROCESS NAME('USER'), INCLUDE, F(I);                          USE00010
 USER: PROC OPTIONS(MAIN);                                     USE00020
                                                               USE00030
 DCL NULL BUILTIN;                                             USE00040
 DCL CONTROL ENTRY EXTERNAL RETURNS(PTR);                      USE00050
 DCL RTN_PTR PTR INIT(NULL());                                 USE00060
 DCL UAP PTR INIT(NULL());                                     USE00070
                                                               USE00080
                                                               USE00090
                                                               USE00100
      RTN_PTR= CONTROL(UAP);                                   USE00110
                                                               USE00120
 END USER;                                                     USE00130
                                                               USE00140
```

```
*PROCESS NAME('ILN'), INCLUDE, F(I);                                    ILN00010
 ILN: PROC (UAP, NODE_NO) RETURNS (BIT);                                ILN00020
 /* ILN= IS_LEVEL2NODE */                                              ILN00030
                                                                        ILN00040
                                                                        ILN00050
      %INCLUDE UA;                                                      ILN00060
      DCL UAP PTR,                                                      ILN00070
          NODE_NO FIXED;                                                ILN00080
                                                                        ILN00090
                                                                        ILN00100
      IF UAP-> UPDN_ARG.NODE_DESCRIP (NODE_NO).PARENT= 0                ILN00110
         THEN RETURN ('1'B);                                            ILN00120
         ELSE RETURN ('0'B);                                            ILN00130
                                                                        ILN00140
                                                                        ILN00150
END ILN;                                                                ILN00160
```

```
*PROCESS NAME('CLN'), INCLUDE, F(I);                                    CLN00010
 CLN: PROC (UAP) RETURNS (FIXED);                                       CLN00020
                                                                        CLN00030
                                                                        CLN00040
 /* CLN= COUNT_LEVEL2NODE */                                            CLN00050
                                                                        CLN00060
                                                                        CLN00070
      %INCLUDE UA;                                                      CLN00080
                                                                        CLN00090
                                                                        CLN00100
      DCL (TRANSLATE, NULL) BUILTIN;                                    CLN00110
      DCL P PTR;                                                        CLN00120
      DCL UAP PTR,                                                      CLN00130
          (NODE_NO, NO_LEVEL2NODE) FIXED INIT(0);                       CLN00140
                                                                        CLN00150
                                                                        CLN00160
      DO NODE_NO= 1 TO UAP-> UPDN_ARG.N;                                CLN00170
         IF UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).PARENT= 0              CLN00180
            THEN NO_LEVEL2NODE= NO_LEVEL2NODE + 1;                      CLN00190
      END;                                                              CLN00200
      RETURN (NO_LEVEL2NODE);                                           CLN00210
                                                                        CLN00220
                                                                        CLN00230
 END CLN;                                                               CLN00240
```

```
*PROCESS NAME('CMN'), INCLUDE, F(I);                                    CMN00010
 /* CMN= COUNT_MTO1NODE */                                              CMN00020
                                                                        CMN00030
                                                                        CMN00040
 CMN: PROC (UAP, BASIC_OP, NO_MTO1VNODE, NO_MTO1NNODE,                  CMN00050
           CP, CT_NO);                                                  CMN00060
 /* CMN= COUNT_MTO1NODE */                                              CMN00070
                                                                        CMN00080
                                                                        CMN00090
      %INCLUDE UA;                                                      CMN00100
      DCL P PTR;                                                        CMN00110
      DCL UAP PTR,                                                      CMN00120
          BASIC_OP CHAR(1),                                             CMN00130
          (NO_MTO1VNODE, NO_MTO1NNODE) FIXED,                           CMN00140
          CP PTR,                                                       CMN00150
          CT_NO FIXED;                                                  CMN00160
      DCL ILN ENTRY EXTERNAL RETURNS(BIT),                              CMN00170
          QCNTYP ENTRY EXTERNAL RETURNS(CHAR(1)),                       CMN00180
          QCEMB ENTRY EXTERNAL RETURNS(BIT);                            CMN00190
      DCL NODE_NO FIXED INIT(0),                                        CMN00200
          BSETID FIXED BIN(15) INIT(0);                                 CMN00210
                                                                        CMN00220
      NO_MTO1VNODE= 0;                                                  CMN00230
      NO_MTO1NNODE= 0;                                                  CMN00240
      DO NODE_NO= 1 TO UAP-> UPDN_ARG.N;                                CMN00250
         IF ILN(UAP, NODE_NO)                                           CMN00260
            THEN IF BASIC_OP= 'M'                                       CMN00270
                    & UAP-> UPDN_ARG.NODE_DESCRIP(NODE_NO).OP= 'I'      CMN00280
                    THEN GOTO EOL;                                      CMN00290
                    ELSE DO;                                            CMN00300
                            BSETID= UAP-> UPDN_ARG.NODE_DESCRIP         CMN00310
                               (NODE_NO).BSETID;                        CMN00320
                            IF QCNTYP(CP, CT_NO, BSETID)= 'V'           CMN00330
                           THEN NO_MTO1VNODE= NO_MTO1VNODE + 1;         CMN00340
                                                                        CMN00350
                            ELSE IF QCEMB(CP, CT_NO, BSETID)            CMN00360
                                        THEN NO_MTO1NNODE=              CMN00370
                                                NO_MTO1NNODE + 1;       CMN00380
                           END;                                         CMN00390
 EOL:                                                                   CMN00400
      END;                                                              CMN00410
                                                                        CMN00420
                                                                        CMN00430
 END CMN;                                                               CMN00440
```

```
*PROCESS NAME('GTRCID'), INCLUDE, F(I);                                  GTR00010
 GTRCID: PROC(UAP, INITIAL_NODE_NO, FINAL_NODE_NO, RECORDID,             GTR00020
                CP, FP, CT_NO)                                           GTR00030
                RETURNS(FIXED);                                          GTR00040
 /* GTRCID= GET_RECORDID */                                              GTR00050
                                                                         GTR00060
                                                                         GTR00070
      %INCLUDE UA;                                                       GTR00080
      %INCLUDE SFL;                                                      GTR00090
                                                                         GTR00100
                                                                         GTR00110
      DCL P PTR;                                                         GTR00120
      DCL QCFLID ENTRY EXTERNAL RETURNS(FIXED BIN(31));                  GTR00130
      DCL QCFDID ENTRY EXTERNAL RETURNS(FIXED BIN(31));                  GTR00140
                                                                         GTR00150
                                                                         GTR00160
      DCL UAP PTR,                                                       GTR00170
          (INITIAL_NODE_NO, FINAL_NODE_NO) FIXED,                        GTR00180
          RECORDID FIXED BIN(31),                                        GTR00190
          (CP, FP) PTR,                                                  GTR00200
          CT_NO FIXED;                                                   GTR00210
                                                                         GTR00220
                                                                         GTR00230
      DCL FILEID FIXED BIN(31) INIT(0),                                  GTR00240
          BSETID FIXED BIN(15) INIT(0),                                  GTR00250
          FIELDID FIXED BIN(31) INIT(0);                                 GTR00260
                                                                         GTR00270
                                                                         GTR00280
      FILEID= QCFLID(CP, CT_NO);                                         GTR00290
      BSETID= UAP-> UPDN_ARG.NODE_DESCRIP(INITIAL_NODE_NO).BSETID;       GTR00300
      FIELDID= QCFDID(CP, CT_NO, BSETID);                                GTR00310
      RECORDID= FP-> FILE(FILEID).RECORD(2).FIELD(FIELDID);              GTR00320
      IF RECORDID=0                                                      GTR00330
         THEN RETURN(1);                                                 GTR00340
         ELSE RETURN(0);                                                 GTR00350
                                                                         GTR00360
                                                                         GTR00370
 END GTRCID;                                                             GTR00380
```

```
*PROCESS NAME('QCEMB'), INCLUDE, F(I);                                  QCE00010
 QCEMB: PROC(CP, CT_NO, BSETID) RETURNS(BIT);                           QCE00020
 /* QCEMB= QUERY EMBEDDED OR NON-EMBEDDED */                           QCE00030
                                                                        QCE00040
                                                                        QCE00050
      %INCLUDE SPCT;                                                    QCE00060
                                                                        QCE00070
                                                                        QCE00080
      DCL CP PTR,                                                       QCE00090
          CT_NO FIXED,                                                  QCE00100
          BSETID FIXED BIN(15);                                         QCE00110
      DCL BD_CTR FIXED INIT(0);                                         QCE00120
                                                                        QCE00130
      DO BD_CTR= 1 TO CP-> PCAT(CT_NO).N;                               QCE00140
         IF CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).BSETID= BSETID        QCE00150
            THEN IF CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).IMP.ITYPE= 'E'QCE00160
                    THEN RETURN('1'B);                                  QCE00170
                    ELSE RETURN('0'B);                                  QCE00180
      END;                                                              QCE00190
                                                                        QCE00200
      RETURN('0'B);                                                     QCE00210
                                                                        QCE00220
 END QCEMB;                                                             QCE00230
```

```
*PROCESS NAME('QCFDID'), INCLUDE, F(I);                                 QCF00010
 QCFDID: PROC(CP, CT_NO, BSETID) RETURNS(FIXED BIN(31));                QCF00020
                                                                        QCF00030
                                                                        QCF00040
 /* QCFDID= QUERY_FIELDID */                                            QCF00050
                                                                        QCF00060
                                                                        QCF00070
                                                                        QCF00080
      %INCLUDE SPCT;                                                    QCF00090
                                                                        QCF00100
                                                                        QCF00110
      DCL P PTR;                                                        QCF00120
      DCL CP PTR,                                                       QCF00130
          CT_NO FIXED,                                                  QCF00140
          BSETID FIXED BIN(15);                                         QCF00150
      DCL BD_CTR FIXED INIT(0);                                         QCF00160
                                                                        QCF00170
      DO BD_CTR= 1 TO CP-> PCAT(CT_NO).N;                               QCF00180
         IF CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).BSETID = BSETID       QCF00190
            THEN RETURN(CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).          QCF00200
                        IMP.FIELDID);                                   QCF00210
      END;                                                              QCF00220
      RETURN (0);                                                       QCF00230
                                                                        QCF00240
                                                                        QCF00250
 END QCFDID;                                                            QCF00260
```

```
*PROCESS NAME('QCFLDS'), INCLUDE, F(I);                          QCF00010
 QCFLDS: PROC(CP, CT_NO, BSETID) RETURNS(FIXED BIN(31));         QCF00020
                                                                 QCF00030
                                                                 QCF00040
 /* QCFLDS= QUERY_FILEID_FOR_SECONDARY_TWOLEV_TREE */            QCF00050
                                                                 QCF00060
                                                                 QCF00070
      %INCLUDE SPCT;                                             QCF00080
                                                                 QCF00090
                                                                 QCF00100
      DCL P PTR;                                                 QCF00110
      DCL CP PTR,                                                QCF00120
          CT_NO FIXED,                                           QCF00130
          BSETID FIXED BIN(15);                                  QCF00140
      DCL BD_CTR FIXED INIT(0);                                  QCF00150
                                                                 QCF00160
                                                                 QCF00170
      DO BD_CTR= 1 TO CP-> PCAT(CT_NO).N;                        QCF00180
         IF CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).BSETID= BSETID QCF00190
            THEN RETURN(CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).   QCF00200
                        IMP.FILEID);                             QCF00210
      END;                                                       QCF00220
      RETURN(0);                                                 QCF00230
                                                                 QCF00240
 END QCFLDS;                                                     QCF00250
```

```
*PROCESS NAME('QCFLID'), INCLUDE, F(I);                          QCF00010
 QCFLID: PROC(CP, CT_NO) RETURNS(FIXED BIN(31));                 QCF00020
                                                                 QCF00030
                                                                 QCF00040
 /* QCFLID= QUERY_FILEID */                                      QCF00050
                                                                 QCF00060
                                                                 QCF00070
      %INCLUDE SPCT;                                             QCF00080
                                                                 QCF00090
                                                                 QCF00100
      DCL P PTR;                                                 QCF00110
      DCL CP PTR,                                                QCF00120
          CT_NO FIXED;                                           QCF00130
      RETURN(CP-> PCAT(CT_NO).FILEID);                           QCF00140
                                                                 QCF00150
                                                                 QCF00160
 END QCFLID;                                                     QCF00170
```

```
*PROCESS NAME('QCNTYP'), INCLUDE, F(I);                                 QCN00010
 QCNTYP: PROC(CP, CT_NO, BSETID) RETURNS(CHAR(1));                      QCN00020
                                                                        QCN00030
                                                                        QCN00040
 /* QCNTYP= QUERY_NODE_TYPE */                                          QCN00050
                                                                        QCN00060
                                                                        QCN00070
      %INCLUDE SPCT;                                                    QCN00080
                                                                        QCN00090
                                                                        QCN00100
      DCL P PTR;                                                        QCN00110
      DCL CP PTR,                                                       QCN00120
          CT_NO FIXED,                                                  QCN00130
          BSETID FIXED BIN(15);                                         QCN00140
      DCL BD_CTR FIXED INIT(0);                                         QCN00150
                                                                        QCN00160
                                                                        QCN00170
      DO BD_CTR= 1 TO  CP->PCAT(CT_NO).N;                               QCN00180
         IF CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).BSETID= BSETID        QCN00190
            THEN RETURN(CP-> PCAT(CT_NO).BSET_DESCRIP(BD_CTR).TYPE);    QCN00200
      END;                                                              QCN00210
      RETURN(' ');                                                      QCN00220
                                                                        QCN00230
                                                                        QCN00240
 END QCNTYP;                                                            QCN00250
```

```
*PROCESS NAME('SBDUA'), INCLUDE, F(I);                                  SBD00010
 SBDUA: PROC RETURNS (PTR);                                             SBD00020
                                                                        SBD00030
                                                                        SBD00040
 /* SBDUA= BUILD_UPDN_ARG */                                            SBD00050
                                                                        SBD00060
                                                                        SBD00070
      %INCLUDE UA;                                                      SBD00080
                                                                        SBD00090
                                                                        SBD00100
      DCL TRANSLATE BUILTIN;                                            SBD00110
      DCL NULL BUILTIN;                                                 SBD00120
      DCL P PTR;                                                        SBD00130
      DCL UA_CTR FIXED INIT(0),                                         SBD00140
          UAP PTR INIT(NULL()),                                         SBD00150
          ND_CTR FIXED INIT(0),                                         SBD00160
          (LOW_BASIC_OP, LOW_OP) CHAR(1);                               SBD00170
                                                                        SBD00180
      PUT SKIP(2) LIST('TOTAL NUMBER OF NON ROOT NODES, FIXED');        SBD00190
      GET LIST(UA_CTR);                                                 SBD00200
      ALLOCATE UPDN_ARG SET (UAP);                                      SBD00210
      UAP-> UPDN_ARG.BASIC_OP= ' ';                                     SBD00220
      UAP-> UPDN_ARG.ROOT_PSETID= 0;                                    SBD00230
      UAP-> UPDN_ARG.NODE_DESCRIP(*).BSETID= 0;                         SBD00240
      UAP-> UPDN_ARG.NODE_DESCRIP(*).PARENT= 0;                         SBD00250
      UAP-> UPDN_ARG.NODE_DESCRIP(*).OP= ' ';                           SBD00260
      UAP-> UPDN_ARG.NODE_DESCRIP(*).DATA= ' ';                         SBD00270
                                                                        SBD00280
                                                                        SBD00290
      GET LIST(LOW_BASIC_OP, UAP-> UPDN_ARG.ROOT_PSETID);               SBD00300
 UAP-> UPDN_ARG.BASIC_OP= TRANSLATE(LOW_BASIC_OP, 'CDM', 'cdm');        SBD00310
                                                                        SBD00320
      DO ND_CTR= 1 TO UA_CTR;                                           SBD00330
         GET LIST(UAP-> UPDN_ARG.NODE_DESCRIP(ND_CTR).BSETID,           SBD00340
                  UAP-> UPDN_ARG.NODE_DESCRIP(ND_CTR).PARENT,           SBD00350
                  LOW_OP,                                               SBD00360
                  UAP-> UPDN_ARG.NODE_DESCRIP(ND_CTR).DATA);            SBD00370
 UAP-> UPDN_ARG.NODE_DESCRIP(ND_CTR).OP                                 SBD00380
   = TRANSLATE(LOW_OP, 'IPRA', 'ipra');                                 SBD00390
      END;                                                              SBD00400
                                                                        SBD00410
      RETURN (UAP);                                                     SBD00420
                                                                        SBD00430
                                                                        SBD00440
 END SBDUA;                                                             SBD00450
```

```
*PROCESS NAME('SFLAPC'), INCLUDE, F(I);                                SFL00010
 SFLAPC: PROC(FP, CP);                                                 SFL00020
                                                                       SFL00030
                                                                       SFL00040
 /* SFLAPC= FILES AND PSET_CATALOGUE */                                SFL00050
                                                                       SFL00060
                                                                       SFL00070
      %INCLUDE SFL;                                                    SFL00080
      %INCLUDE SPCT;                                                   SFL00090
                                                                       SFL00100
                                                                       SFL00110
      DCL P PTR;                                                       SFL00120
      DCL (FP, CP) PTR;                                                SFL00130
                                                                       SFL00140
                                                                       SFL00150
      ALLOCATE FILE SET (FP);                                          SFL00160
      ALLOCATE PCAT SET (CP);                                          SFL00170
                                                                       SFL00180
      CP-> PCAT(*).PSETID= 0;                                          SFL00190
      CP-> PCAT(*).FILEID= 0;                                          SFL00200
      CP-> PCAT(*).BSET_DESCRIP(*).BSETID= 0;                          SFL00210
      CP-> PCAT(*).BSET_DESCRIP(*).TYPE= ' ';                          SFL00220
      CP-> PCAT(*).BSET_DESCRIP(*).FUNC= ' ';                          SFL00230
      CP-> PCAT(*).BSET_DESCRIP(*).IMP.ITYPE= ' ';                     SFL00240
      CP-> PCAT(*).BSET_DESCRIP(*).IMP.FIELDID=0;                      SFL00250
      CP-> PCAT(*).BSET_DESCRIP(*).IMP.FILEID=0;                       SFL00260
                                                                       SFL00270
                                                                       SFL00280
      CP-> PCAT(3).PSETID= 10000;                                      SFL00290
      CP-> PCAT(3).FILEID= 10;                                         SFL00300
      CP-> PCAT(3).N= 5;                                               SFL00310
      CP-> PCAT(3).BSET_DESCRIP(1).BSETID= 304;                        SFL00320
      CP-> PCAT(3).BSET_DESCRIP(1).TYPE= 'V';                          SFL00330
      CP-> PCAT(3).BSET_DESCRIP(1).FUNC= '1';                          SFL00340
      CP-> PCAT(3).BSET_DESCRIP(1).IMP.ITYPE= 'E';                     SFL00350
      CP-> PCAT(3).BSET_DESCRIP(1).IMP.FIELDID= 1;                     SFL00360
      CP-> PCAT(3).BSET_DESCRIP(2).BSETID= 303;                        SFL00370
      CP-> PCAT(3).BSET_DESCRIP(2).TYPE= 'V';                          SFL00380
      CP-> PCAT(3).BSET_DESCRIP(2).FUNC= '1';                          SFL00390
      CP-> PCAT(3).BSET_DESCRIP(2).IMP.ITYPE= 'E';                     SFL00400
      CP-> PCAT(3).BSET_DESCRIP(2).IMP.FIELDID= 2;                     SFL00410
      CP-> PCAT(3).BSET_DESCRIP(3).BSETID= 302;                        SFL00420
      CP-> PCAT(3).BSET_DESCRIP(3).TYPE= 'N';                          SFL00430
      CP-> PCAT(3).BSET_DESCRIP(3).FUNC= 'M';                          SFL00440
      CP-> PCAT(3).BSET_DESCRIP(3).IMP.ITYPE= 'E';                     SFL00450
      CP-> PCAT(3).BSET_DESCRIP(3).IMP.FIELDID= 3;                     SFL00460
      CP-> PCAT(3).BSET_DESCRIP(4).BSETID= 102;                        SFL00470
      CP-> PCAT(3).BSET_DESCRIP(4).TYPE= 'N';                          SFL00480
      CP-> PCAT(3).BSET_DESCRIP(4).FUNC= '0';                          SFL00490
      CP-> PCAT(3).BSET_DESCRIP(4).IMP.ITYPE= 'N';                     SFL00500
      CP-> PCAT(3).BSET_DESCRIP(4).IMP.FIELDID= 1;                     SFL00510
      CP-> PCAT(3).BSET_DESCRIP(4).IMP.FILEID= 8;                      SFL00520
      CP-> PCAT(3).BSET_DESCRIP(5).BSETID= 401;                        SFL00530
      CP-> PCAT(3).BSET_DESCRIP(5).TYPE= 'N';                          SFL00540
      CP-> PCAT(3).BSET_DESCRIP(5).FUNC= '0';                          SFL00550
```

```
CP-> PCAT(3).BSET_DESCRIP(5).IMP.ITYPE= 'N';                    SFL00560
CP-> PCAT(3).BSET_DESCRIP(5).IMP.FIELDID= 2;                    SFL00570
CP-> PCAT(3).BSET_DESCRIP(5).IMP.FILEID= 7;                     SFL00580
                                                                SFL00590
                                                                SFL00600
CP-> PCAT(2).PSETID= 20000;                                     SFL00610
CP-> PCAT(2).FILEID= 9;                                         SFL00620
CP-> PCAT(2).N= 5;                                              SFL00630
CP-> PCAT(2).BSET_DESCRIP(1).BSETID= 201;                       SFL00640
CP-> PCAT(2).BSET_DESCRIP(1).TYPE= 'V';                         SFL00650
CP-> PCAT(2).BSET_DESCRIP(1).FUNC= '1';                         SFL00660
CP-> PCAT(2).BSET_DESCRIP(1).IMP.ITYPE= 'E';                    SFL00670
CP-> PCAT(2).BSET_DESCRIP(1).IMP.FIELDID= 2;                    SFL00680
CP-> PCAT(2).BSET_DESCRIP(2).BSETID= 202;                       SFL00690
CP-> PCAT(2).BSET_DESCRIP(2).TYPE= 'V';                         SFL00700
CP-> PCAT(2).BSET_DESCRIP(2).FUNC= '1';                         SFL00710
CP-> PCAT(2).BSET_DESCRIP(2).IMP.ITYPE= 'E';                    SFL00720
CP-> PCAT(2).BSET_DESCRIP(2).IMP.FIELDID= 1;                    SFL00730
CP-> PCAT(2).BSET_DESCRIP(3).BSETID= 302;                       SFL00740
CP-> PCAT(2).BSET_DESCRIP(3).TYPE= 'N';                         SFL00750
CP-> PCAT(2).BSET_DESCRIP(3).FUNC= '0';                         SFL00760
CP-> PCAT(2).BSET_DESCRIP(3).IMP.ITYPE= 'N';                    SFL00770
CP-> PCAT(2).BSET_DESCRIP(3).IMP.FIELDID= 3;                    SFL00780
CP-> PCAT(2).BSET_DESCRIP(3).IMP.FILEID= 10;                    SFL00790
CP-> PCAT(2).BSET_DESCRIP(4).BSETID= 203;                       SFL00800
CP-> PCAT(2).BSET_DESCRIP(4).TYPE= 'V';                         SFL00810
CP-> PCAT(2).BSET_DESCRIP(4).FUNC= '1';                         SFL00820
CP-> PCAT(2).BSET_DESCRIP(4).IMP.ITYPE= 'E';                    SFL00830
CP-> PCAT(2).BSET_DESCRIP(4).IMP.FIELDID= 3;                    SFL00840
                                                                SFL00850
                                                                SFL00860
CP-> PCAT(1).PSETID= 30000;                                     SFL00870
CP-> PCAT(1).FILEID= 8;                                         SFL00880
CP-> PCAT(1).N= 5;                                              SFL00890
CP-> PCAT(1).BSET_DESCRIP(1).BSETID= 101;                       SFL00900
CP-> PCAT(1).BSET_DESCRIP(1).TYPE= 'V';                         SFL00910
CP-> PCAT(1).BSET_DESCRIP(1).FUNC= '1';                         SFL00920
CP-> PCAT(1).BSET_DESCRIP(1).IMP.ITYPE= 'E';                    SFL00930
CP-> PCAT(1).BSET_DESCRIP(1).IMP.FIELDID= 2;                    SFL00940
CP-> PCAT(1).BSET_DESCRIP(2).BSETID= 102;                       SFL00950
CP-> PCAT(1).BSET_DESCRIP(2).TYPE= 'N';                         SFL00960
CP-> PCAT(1).BSET_DESCRIP(2).FUNC= 'M';                         SFL00970
CP-> PCAT(1).BSET_DESCRIP(2).IMP.ITYPE= 'E';                    SFL00980
CP-> PCAT(1).BSET_DESCRIP(2).IMP.FIELDID= 1;                    SFL00990
CP-> PCAT(1).BSET_DESCRIP(3).BSETID= 103;                       SFL01000
CP-> PCAT(1).BSET_DESCRIP(3).TYPE= 'V';                         SFL01010
CP-> PCAT(1).BSET_DESCRIP(3).FUNC= '1';                         SFL01020
CP-> PCAT(1).BSET_DESCRIP(3).IMP.ITYPE= 'E';                    SFL01030
CP-> PCAT(1).BSET_DESCRIP(3).IMP.FIELDID= 3;                    SFL01040
CP-> PCAT(1).BSET_DESCRIP(4).BSETID= 104;                       SFL01050
CP-> PCAT(1).BSET_DESCRIP(4).TYPE= 'V';                         SFL01060
CP-> PCAT(1).BSET_DESCRIP(4).FUNC= '1';                         SFL01070
CP-> PCAT(1).BSET_DESCRIP(4).IMP.ITYPE= 'E';                    SFL01080
CP-> PCAT(1).BSET_DESCRIP(4).IMP.FIELDID= 4;                    SFL01090
                                                                SFL01100
```

```
                                                                    SFLO1110
                                                                    SFLO1120
CP-> PCAT(4).PSETID= 25000;                                         SFLO1130
CP-> PCAT(4).FILEID= 7;                                             SFLO1140
CP-> PCAT(4).N= 5;                                                  SFLO1150
CP-> PCAT(4).BSET_DESCRIP(1).BSETID= 401;                           SFLO1160
CP-> PCAT(4).BSET_DESCRIP(1).TYPE= 'N';                             SFLO1170
CP-> PCAT(4).BSET_DESCRIP(1).FUNC= 'M';                             SFLO1180
CP-> PCAT(4).BSET_DESCRIP(1).IMP.ITYPE= 'E';                        SFLO1190
CP-> PCAT(4).BSET_DESCRIP(1).IMP.FIELDID= 2;                        SFLO1200
CP-> PCAT(4).BSET_DESCRIP(2).BSETID= 402;                           SFLO1210
CP-> PCAT(4).BSET_DESCRIP(2).TYPE= 'V';                             SFLO1220
CP-> PCAT(4).BSET_DESCRIP(2).FUNC= '1';                             SFLO1230
CP-> PCAT(4).BSET_DESCRIP(2).IMP.ITYPE= 'E';                        SFLO1240
CP-> PCAT(4).BSET_DESCRIP(2).IMP.FIELDID= 1;                        SFLO1250
                                                                    SFLO1260
                                                                    SFLO1270
CP-> PCAT(4).BSET_DESCRIP(3).BSETID= 403;                           SFLO1280
CP-> PCAT(4).BSET_DESCRIP(3).TYPE= 'N';                             SFLO1290
CP-> PCAT(4).BSET_DESCRIP(3).FUNC= 'M';                             SFLO1300
CP-> PCAT(4).BSET_DESCRIP(3).IMP.ITYPE= 'E';                        SFLO1310
CP-> PCAT(4).BSET_DESCRIP(3).IMP.FIELDID= 3;                        SFLO1320
                                                                    SFLO1330
                                                                    SFLO1340
CP-> PCAT(5).PSETID= 15000;                                         SFLO1350
CP-> PCAT(5).FILEID= 6;                                             SFLO1360
CP-> PCAT(5).N= 5;                                                  SFLO1370
CP-> PCAT(5).BSET_DESCRIP(1).BSETID= 403;                           SFLO1380
CP-> PCAT(5).BSET_DESCRIP(1).TYPE= 'N';                             SFLO1390
CP-> PCAT(5).BSET_DESCRIP(1).FUNC= 'O';                             SFLO1400
CP-> PCAT(5).BSET_DESCRIP(1).IMP.ITYPE= 'N';                        SFLO1410
CP-> PCAT(5).BSET_DESCRIP(1).IMP.FIELDID= 3;                        SFLO1420
CP-> PCAT(5).BSET_DESCRIP(1).IMP.FILEID= 7;                         SFLO1430
CP-> PCAT(5).BSET_DESCRIP(2).BSETID= 501;                           SFLO1440
CP-> PCAT(5).BSET_DESCRIP(2).TYPE= 'V';                             SFLO1450
CP-> PCAT(5).BSET_DESCRIP(2).FUNC= '1';                             SFLO1460
CP-> PCAT(5).BSET_DESCRIP(2).IMP.ITYPE= 'E';                        SFLO1470
CP-> PCAT(5).BSET_DESCRIP(2).IMP.FIELDID= 1;                        SFLO1480
CP-> PCAT(5).BSET_DESCRIP(3).BSETID= 502;                           SFLO1490
CP-> PCAT(5).BSET_DESCRIP(3).TYPE= 'V';                             SFLO1500
CP-> PCAT(5).BSET_DESCRIP(3).FUNC= '1';                             SFLO1510
CP-> PCAT(5).BSET_DESCRIP(3).IMP.ITYPE= 'E';                        SFLO1520
CP-> PCAT(5).BSET_DESCRIP(3).IMP.FIELDID= 3;                        SFLO1530
                                                                    SFLO1540
                                                                    SFLO1550
                                                                    SFLO1560
                                                                    SFLO1570
                                                                    SFLO1580
                                                                    SFLO1590
                                                                    SFLO1600
                                                                    SFLO1610
                                                                    SFLO1620
FP-> FILE(*).RECORD(*).FIELD(*)= 0;                                 SFLO1630
                                                                    SFLO1640
                                                                    SFLO1650
```

```
      FP-> FILE(10).RECORD(2).FIELD(1)= 1000;                    SFLO1660
      FP-> FILE(10).RECORD(2).FIELD(2)= 21;                      SFLO1670
      FP-> FILE(10).RECORD(2).FIELD(3)= 2;                       SFLO1680
      FP-> FILE(9).RECORD(2).FIELD(1)= 15;                       SFLO1690
      FP-> FILE(9).RECORD(2).FIELD(2)= 300;                      SFLO1700
      FP-> FILE(9).RECORD(2).FIELD(3)= 30;                       SFLO1710
      FP-> FILE(8).RECORD(2).FIELD(1)= 2;                        SFLO1720
      FP-> FILE(8).RECORD(2).FIELD(2)= 1;                        SFLO1730
      FP-> FILE(8).RECORD(2).FIELD(3)= 3;                        SFLO1740
      FP-> FILE(8).RECORD(2).FIELD(4)= 2000;                     SFLO1750
      FP-> FILE(7).RECORD(2).FIELD(1)= 37;                       SFLO1760
      FP-> FILE(7).RECORD(2).FIELD(2)= 2;                        SFLO1770
                                                                 SFLO1780
      FP-> FILE(7).RECORD(2).FIELD(3)= 2;                        SFLO1790
      FP-> FILE(6).RECORD(2).FIELD(1)= 1369;                     SFLO1800
      FP-> FILE(6).RECORD(2).FIELD(3)= 41;                       SFLO1810
                                                                 SFLO1820
END SFLAPC;                                                      SFLO1830
```

Appendix 2- Data Structures

The following PL/1 data structures are included in this appendix in order.

UPDN_ARG

UPDN_RTN

TWOLEV_TREE

MTO1VNODE_INFO

MTO1NNODE_INFO

OTOMNODE_INFO

PTR_ARR

PCAT

FILE

```
DCL 1 UPDN_ARG BASED (P),                                    UA 00010
    2 BASIC_OP CHAR(1),                                      UA 00020
    2 ROOT_PSETID FIXED BIN(15),                             UA 00030
    2 N FIXED,                                               UA 00040
    2 NODE_DESCRIP(UA_CTR REFER (UPDN_ARG.N)),               UA 00050
      3 BSETID FIXED BIN(15),                                UA 00060
      3 PARENT FIXED,                                        UA 00070
      3 OP CHAR(1),                                          UA 00080
      3 DATA CHAR(40) VAR;                                   UA 00090
```

```
DCL 1 UPDN_RTN BASED(P),                                    UR 00010
    2 RTN_CODE FIXED,                                       UR 00020
    2 ERROR_NODENO FIXED;                                   UR 00030
```

```
DCL 1 TWOLEV_TREE BASED (P),                                    TW000010
      2 BASIC_OP CHAR (1),                                      TW000020
      2 B_OP_RTNCODE FIXED,                                     TW000030
      2 ROOT_FILEID FIXED BIN (31),                             TW000040
      2 RECORDID FIXED BIN (31),                                TW000050
      2 NO_OF_FIELD FIXED,                                      TW000060
      2 ID_ARR (NO_FDUPD REFER (TWOLEV_TREE.NO_OF_FIELD)),      TW000070
        3 FIELDID FIXED BIN (31),                               TW000080
        3 DATA CHAR (40) VAR;                                   TW000090
```

```
DCL 1 MTO1VNODE_INFO BASED (P),                                          MVN00010
      2 NO_MTO1VNODE FIXED,                                              MVN00020
      2 INFO_ARR (VIA_CTR REFER (MTO1VNODE_INFO.NO_MTO1VNODE)),          MVN00030
        3 NODE_NO FIXED,                                                 MVN00040
        3 FIELDID FIXED BIN(31),                                         MVN00050
        3 DATA CHAR (40) VAR;                                            MVN00060
```

```
DCL 1 MTO1NNODE_INFO BASED (P),                                    MNN00010
      2 NO_MTO1NNODE FIXED,                                        MNN00020
      2 INFO_ARR (NIA_CTR REFER (MTO1NNODE_INFO.NO_MTO1NNODE)),    MNN00030
        3 INITIAL_NODE_NO FIXED,                                   MNN00040
        3 FINAL_NODE_NO FIXED,                                     MNN00050
        3 FIELDID FIXED BIN(31);                                   MNN00060
```

```
DCL 1 OTOMNODE_INFO BASED (P),                                          ONI00010
      2 NO_1TOMNODE FIXED,                                              ONI00020
      2 INFO_ARR (ONIA_CTR REFER (OTOMNODE_INFO.NO_1TOMNODE)) FIXED;    ONI00030
```

```
DCL 1 PTR_ARR BASED (P),                                        PA 00010
      2 NO_PTR FIXED,                                           PA 00020
      2 ARR(PA_CTR REFER(PTR_ARR.NO_PTR)) PTR;                  PA 00030
```

```
DCL 1 PCAT(5) BASED (P),                                    SPC00010
      2 PSETID FIXED BIN(15),                               SPC00020
      2 FILEID FIXED BIN(31),                               SPC00030
      2 N FIXED,                                            SPC00040
      2 BSET_DESCRIP (5),                                   SPC00050
        3 BSETID FIXED BIN(15),                             SPC00060
        3 TYPE CHAR(1),                                     SPC00070
        3 FUNC CHAR(1),                                     SPC00080
        3 IMP,                                              SPC00090
          4 ITYPE CHAR(1),                                  SPC00100
          4 FIELDID FIXED BIN(31),                          SPC00110
          4 FILEID FIXED BIN(31);                           SPC00120
```

```
DCL 1 FILE (10) BASED (P),                                    SFL00010
    2 RECORD (2),                                             SFL00020
      3 FIELD (4) FIXED;                                      SFL00030
```

Appendix 3- Sample Terminal Session

The following are a few testings with all the three types of trees- create, delete and modify. The complete testing can be accomplished by running through the following exec files:

ex31 to ex40 for testing create

ex51 to ex52 for testing delete

ex71 to ex92 for testing modify

```
R; T=0.01/0.01 12:06:44
ex32
R; T=0.01/0.02 12:06:57
EXECUTION BEGINS...

TOTAL NUMBER OF NON ROOT NODES, FIXED
:
:
:
:
NUMBER OF LEVEL 2 NODES IS                          2
NUMBER OF MTO1 VALUE NODES IS                       2
NUMBER OF MTO1 ENTITY NODES IS                      0
PRINTING CONTENTS OF TWOLEV_TREE
TLT.BASIC_OP              C
TLT.ROOT_FILEID                     9
ENTRY                          1              :                 FIELDID AND DATA
            2          300
ENTRY                          2              :                 FIELDID AND DATA
            1          15
ENTRY                          3              :                 FIELDID AND DATA
            3          NULL
PRINTING CONTENTS OF MVNI
ENTRY                          1              :                 NODE_NO, FIELDID, DATA
      1                              1          15
ENTRY                          2              :                 NODE_NO, FIELDID, DATA
      2                              2          300
PRINTING CONTENTS OF TWOLEV_TREE
ENTRY                          1              :                 FIELDID AND DATA
            2          300
ENTRY                          2              :                 FIELDID AND DATA
            1          15
ENTRY                          3              :                 FIELDID AND DATA
            3          NULL
PRIMARY TREE PROCESSED
OPERATION FOR MTO1 V AND N NODES SUCCESSFUL
BASIC_OP RTNCODE AND ERROR_NODENO
      0                        0
OPERATION COMPLETELY SUCCESSFUL
BASIC_OP RETNCODE, ERROR_NODENO
      0                        0
R; T=0.68/1.45 12:07:43
ex37
R; T=0.01/0.01 12:07:50
EXECUTION BEGINS...

TOTAL NUMBER OF NON ROOT NODES, FIXED
:
:
:
:
NUMBER OF LEVEL 2 NODES IS                          2
NUMBER OF MTO1 VALUE NODES IS                       1
NUMBER OF MTO1 ENTITY NODES IS                      1
PRINTING CONTENTS OF TWOLEV_TREE
TLT.BASIC_OP              C
TLT.ROOT_FILEID                     8
ENTRY                          1              :                 FIELDID AND DATA
            2          NULL
```

```
ENTRY                          2                :          FIELDID AND DATA
          1          NULL
ENTRY                          3                :          FIELDID AND DATA
          3          3
ENTRY                          4                :          FIELDID AND DATA
          4          NULL
PRINTING CONTENTS OF MVNI
ENTRY                          1                :          NODE_NO, FIELDID, DATA
      3                              3          3
PRINTING CONTENTS OF MNNI
ENTRY                          1                :          INITIAL_NODE_NO, FIELDID
      1                              1
PRINTING CONTENTS OF TWOLEV_TREE
ENTRY                          1                :          FIELDID AND DATA
          2          NULL
ENTRY                          2                :          FIELDID AND DATA
          1                    2
ENTRY                          3                :          FIELDID AND DATA
          3          3
ENTRY                          4                :          FIELDID AND DATA
          4          NULL
PRINTING CONTENTS OF MNNI
ENTRY                          1                :          INITIAL_NODE_NO, FINAL_NODE_NO, FIELDID
      1                        3                      1
PRIMARY TREE PROCESSED
OPERATION FOR MTO1 V AND N NODES SUCCESSFUL
BASIC_OP RTNCODE AND ERROR_NODENO
      0                        0
OPERATION COMPLETELY SUCCESSFUL
BASIC_OP RETNCODE, ERROR_NODENO
      0                        0
R; T=0.68/1.38 12:08:06
ex51
R; T=0.01/0.01 12:08:10
EXECUTION BEGINS...

TOTAL NUMBER OF NON ROOT NODES, FIXED
:
:
:
:
:
:
NUMBER OF LEVEL 2 NODES IS                          3
NUMBER OF MTO1 VALUE NODES IS                       2
NUMBER OF MTO1 ENTITY NODES IS                      0
PRINTING CONTENTS OF TWOLEV_TREE
TLT.BASIC_OP            D
TLT.ROOT_FILEID                    9
PRINTING CONTENTS OF TWOLEV_TREE
RECORDID OF ROOT_NODE
          300
OPERATION COMPLETELY SUCCESSFUL
BASIC_OP RETNCODE, ERROR_NODENO
      0                        0
R; T=0.64/1.26 12:08:28
ex52
R; T=0.01/0.01 12:08:36
EXECUTION BEGINS...
```

```
TOTAL NUMBER OF NON ROOT NODES, FIXED
:
:
:
:
:
NUMBER OF LEVEL 2 NODES IS                          3
NUMBER OF MTO1 VALUE NODES IS                       3
NUMBER OF MTO1 ENTITY NODES IS                      0
PRINTING CONTENTS OF TWOLEV_TREE
TLT.BASIC_OP            D
TLT.ROOT_FILEID                    8
PRINTING CONTENTS OF TWOLEV_TREE
RECORDID OF ROOT_NODE
          3
OPERATION COMPLETELY SUCCESSFUL
BASIC_OP RETNCODE, ERROR_NODENO
     0                     0
R; T=0.64/1.29 12:08:51
ex72
R; T=0.01/0.02 12:09:09
EXECUTION BEGINS...

TOTAL NUMBER OF NON ROOT NODES, FIXED
:
:
:
:
:
NUMBER OF LEVEL 2 NODES IS                          3
NUMBER OF MTO1 VALUE NODES IS                       2
NUMBER OF MTO1 ENTITY NODES IS                      0
PRINTING CONTENTS OF TWOLEV_TREE
TLT.BASIC_OP            M
TLT.ROOT_FILEID                    9
ENTRY                         1              :          FIELDID AND DATA
          1          15
ENTRY                         2              :          FIELDID AND DATA
          2          300
PRINTING CONTENTS OF MVNI
ENTRY                         1              :          NODE_NO, FIELDID, DATA
     2                        1         15
ENTRY                         2              :          NODE_NO, FIELDID, DATA
     3                        2         300
PRINTING CONTENTS OF TWOLEV_TREE
RECORDID OF ROOT_NODE
          30
ENTRY                         1              :          FIELDID AND DATA
          1          15
ENTRY                         2              :          FIELDID AND DATA
          2          300
PRIMARY TREE PROCESSED
OPERATION FOR MTO1 V AND N NODES SUCCESSFUL
BASIC_OP RTNCODE AND ERROR_NODENO
     0                     0
OPERATION COMPLETELY SUCCESSFUL
BASIC_OP RETNCODE, ERROR_NODENO
     0                     0
R; T=0.67/1.39 12:09:27
ex82
```

```
R; T=0.01/0.02 12:09:31
EXECUTION BEGINS...

TOTAL NUMBER OF NON ROOT NODES, FIXED
:
:
:
:
:
:
:
:
:
NUMBER OF LEVEL 2 NODES IS                          3
NUMBER OF MTO1 VALUE NODES IS                       0
NUMBER OF MTO1 ENTITY NODES IS                      0
PRINTING CONTENTS OF TWOLEV_TREE
TLT.BASIC_OP              M
TLT.ROOT_FILEID                    10
ENTRY                         1              :              FIELDID AND DATA
          0
PRINTING CONTENTS OF ONI
ENTRY                         1              :              NODE_NO
      3
ENTRY                         2              :              NODE_NO
      6
PRINTING CONTENTS OF TWOLEV_TREE
RECORDID OF ROOT_NODE
          2
ENTRY                         1              :              FIELDID AND DATA
          0
PRINTING CONTENTS OF PTR_ARR
ENTRY                         1              :              BASIC_OP, FILEID
M                                    7
RECORDID, NO_FDUPD
          21                  1
FIELDID, DATA
          2                          2
ENTRY                         2              :              BASIC_OP, FILEID
M                                    8
RECORDID, NO_FDUPD
         1000                 1
FIELDID, DATA
          1                          2
NO PRIMARY TREE PROCESSED
OPERATION FOR MTO1 V AND N NODES SUCCESSFUL
BASIC_OP RTNCODE AND ERROR_NODENO
      0                      0
OPERATION COMPLETELY SUCCESSFUL
BASIC_OP RETNCODE, ERROR_NODENO
      0                      0
R; T=0.69/1.42 12:09:53
cp sp cons stop close
```