This paper describes a SAS procedure, written in PL/I, that was developed as an update machine for the Consumer Price and Consumption Studies (CPCS) Branch of the Bureau of Labor Statistics. This procedure updates a file stored as a relation in the RAPID Data Base Management System. The input transaction records are stored in a SAS data set. The procedure is executed in one of three mutually-exclusive update modes: ADD, for adding rows (records) to a RAPID relation; DELETE, for deleting rows; and CHANGE for modifying the values of variables on existing rows. The procedure is also executed in one of four incremental levels of processing: SCAN, for scanning the user's PROC UPDATE statement; EDIT, for scanning and then editing the user's SAS variables; TEST, for scanning, editing, and accessing RAPID rows; and UPDATE, for scanning, editing, testing and actually updating the RAPID relation. The user specifies the name of the RAPID relation (RELN=), the update mode and the level of processing as the required parameters on the PROC statement. For example:

```
PROC UPDATE RELN=ABCD MODE=ADD PROCESS=TEST;
```

The procedure interfaces to the RAPID environment through the RAPID/PL/I interface software available through RAPID. RAPID commands are coded to search key trees for row identification and to access RAPID rows. The procedure performs additions of rows top-down and deletions bottom-up. Before a row is added, the procedure will ensure that the parent record, if the relation has a parent, does exist. Before a row is deleted, the procedure will ensure that child records, orphans, are not left behind.

Another interface used by this procedure is to the data dictionary maintained by CPCS. This dictionary defines the RAPID data base. At compile time the procedure includes PL/I code from the data dictionary system. This code enables retrieval of dictionary information by means of a hashing function. This interface verifies that the RELN (relation) specified by the user is in the data base. It also obtains the variables in the RAPID relation and the attributes of each variable including its null or default value, editing criteria, alias variable name, and the value to use if it is to be recoded.

The interface to this dictionary is also used to obtain user-defined hierarchical relationships among relations in a RAPID data base. A relation can be defined to be "parent" to another, or a relation can have "child" relations. To maintain the integrity of these relationships, the procedure performs additions of rows top-down and deletions bottom-up. Before a row is added, the procedure will ensure that the parent record, if the relation has a parent, does exist. Before a row is deleted, the procedure will ensure that child records, orphans, are not left behind.

This procedure operates in the SAS environment. DATA and PROC steps can occur both before and after invoking the procedure. A DATA step prior to the procedure, for example, can create or modify the transaction SAS data set used as the procedure's input. SAS datasets containing errors and updates, which are optionally created by the procedure, can be used in subsequent DATA and PROC steps. The full power of SAS is available to these data sets: they can be printed, sorted, sorted permanently, and merged with other SAS data sets. Examples of printing (PROC PRINT) these two data sets are illustrated in Figures 3-5.

```
Figure 2
```

```
/*$STOR FILE (RELN) INDX (WORK ROWINDEX) CLMS (NAMES) FROM (WORK RECORD); */
DO;$
$OP='$STOR';
$FILE=RELN;
CALL $INTER($OP,$STAT, $FILE, WORK ROWINDEX, NAMES, WORK RECORD); 
IF SUBSTR($STAT,1,1)= 'I THEN SIGNAL CONDITION ($ERROR);
END;
```

Figure 1

Source Statements

```
$INCLUDE $PRIM;
/*BRINGS IN NECESSARY RAPID DECLARATIONS AT COMPIL TIME*/
$STOR FILE (RELN) INDX (WORK ROWINDEX) CLMS (NAMES) FROM (WORK RECORD);
```
Figure 3
DATA ABCD;
/* CREATE TRANSACTION SAS DATA SET */
;
PROC UPDATE RELN=ABCD MODE=ADD PROCESS=UPDATE ERROR=ERROR CHANGES=CHANGES DATA=ABCD;
TITLE PROC UPDATE;
PROC PRINT DATA=ERROR;
FORMAT ERROR ERROR.
TITLE ERROR DATASET CREATED BY UPDATE;
PROC PRINT DATA=CHANGES;
TITLE CHANGES DATASET CREATED BY UPDATE;

Figure 4
ERROR DATASET CREATED BY UPDATE ADD MODE

<table>
<thead>
<tr>
<th>OBS</th>
<th>KEY</th>
<th>ERROR</th>
<th>SAS_OBS</th>
<th>MODE</th>
<th>RELN</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000011</td>
<td>INVALID CODE FOR CPC R CODED FIELD TYPE</td>
<td>1</td>
<td>ADD</td>
<td>ABCD</td>
<td>OH</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>0000011</td>
<td>FIXED BINARY VALUE MUST BE AN INTEGER</td>
<td>1</td>
<td>ADD</td>
<td>ABCD</td>
<td>RTN_DATE</td>
<td>2.91</td>
</tr>
<tr>
<td>3</td>
<td>0000011</td>
<td>FIXED BINARY VALUE MUST BE AN INTEGER</td>
<td>2</td>
<td>ADD</td>
<td>ABCD</td>
<td>OH</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0000022</td>
<td>INVALID CODE FOR CPC R CODED FIELD TYPE</td>
<td>3</td>
<td>ADD</td>
<td>ABCD</td>
<td>DT_ABCD</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>0000033</td>
<td>NON-EIGHT CHARACTER IN DATA</td>
<td>4</td>
<td>ADD</td>
<td>ABCD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5
CHANGES DATASET CREATED BY UPDATE ADD MODE

<table>
<thead>
<tr>
<th>OBS</th>
<th>RAPIDRON</th>
<th>KEY</th>
<th>SAS_OBS</th>
<th>MODE</th>
<th>RELN</th>
<th>RECTYPE</th>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0000011</td>
<td>1</td>
<td>ADD</td>
<td>ABCD</td>
<td>COMMON</td>
<td>PRIM KEY</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0000022</td>
<td>2</td>
<td>ADD</td>
<td>ABCD</td>
<td>COMMON</td>
<td>PRIM KEY</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0000033</td>
<td>3</td>
<td>ADD</td>
<td>ABCD</td>
<td>COMMON</td>
<td>PRIM KEY</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0000066</td>
<td>4</td>
<td>ADD</td>
<td>ABCD</td>
<td>COMMON</td>
<td>PRIM KEY</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0000113</td>
<td>5</td>
<td>ADD</td>
<td>ABCD</td>
<td>COMMON</td>
<td>PRIM KEY</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0000124</td>
<td>6</td>
<td>ADD</td>
<td>ABCD</td>
<td>COMMON</td>
<td>PRIM KEY</td>
<td></td>
</tr>
</tbody>
</table>

*PRIM KEY* informs the user that the row was added with the values of the keys. All other variables are loaded with their default values.
The procedure can be invoked repeatedly in a SAS job step. Since the procedure updates only one relation in a RAPID database at a time, repeated invocations can be used to update several RAPID relations in one SAS job step. DELENT, an application program using PROC UPDATE and the MACRO language, invokes the procedure repeatedly to delete hierarchically-related rows in relations of a RAPID database.

In the user's view, the procedure is designed to be consistent with other SAS procedures. DATA=data set name is the means by which the user specifies the input SAS data set. As in other SAS procedures, the default is the last SAS data set created. If not explicitly specified on the VAR (or VARIABLES) statement accompanying the procedure, the SAS transaction variables, as in standard SAS procedures, default to all variables in the input data set that are not on another list.

Messages written to the SAS Log are consistent with those already familiar to the user. They begin with "NOTE:", "WARNING:" or "ERROR:"

Further information about PROC UPDATE is available from:

Linda A. Smith  
ORI, Inc.  
Suite 250  
122 C Street, N.W.  
Washington, D.C. 20001  
202-737-2666

For more information about the data dictionary system contact:

Stephen E. Weiss  
Bureau of Labor Statistics  
600 E Street, N.W.  
Washington, D.C. 20001  
202-272-2553