TCT Structure Changes: Detailed Design

Information Systems & Computing
University of California
Office of the President

September 12, 2000

Table of Contents

• Overview

• Changes to TCT Tables
  • Elements Moved from One Table to Another
  • TCT Tables Dropped
  • TCT Tables Added

• Program Changes
  • One-Time Program for TCT Conversion
  • TCT Update Process
    • Forms/Transaction Changes
    • Update Programs and Components
  • Changes to the Title Code Table Utility, PPTCTUTL
  • Programs Converted to Use PPTCTUTL for TCT Access
Overview

This document is prepared in response to the requirements for changes to the structure of the Payroll/Personnel System's (PPS) repository of title-related information, the Title Code Table (TCT). Those requirements accompanied service request 17315 and were themselves prepared in response to the changes which are being applied to the University Title Code System (TCS).

This document will describe in detail the structure of the tables in a revised TCT. Changes to transactions and forms used in updating the TCT will also be treated in detail. Changes to PPS programs and processes will be described in general terms.
Changes to TCT Tables

An analysis of the requirements definition, the changes which are being made to TCS and the current patterns of usage of TCT data by PPS have led to a redesign of TCT. Graphical representations of the structures involved in this project are useful in understanding the starting point in PPS and the target which is set by the new TCT. The following table contains links to diagrams of these structures:

<table>
<thead>
<tr>
<th>Database</th>
<th>Diagram Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current PPS Title Code Table</td>
<td><a href="http://www.ucop.edu/~wgayle/tctmap.html">http://www.ucop.edu/~wgayle/tctmap.html</a></td>
</tr>
<tr>
<td>Changing TCS Table Structure</td>
<td><a href="http://www.ucop.edu/~wgayle/tcsmap.html">http://www.ucop.edu/~wgayle/tcsmap.html</a></td>
</tr>
<tr>
<td>New PPS Title Code Table</td>
<td><a href="http://www.ucop.edu/~wgayle/newtct.html">http://www.ucop.edu/~wgayle/newtct.html</a></td>
</tr>
<tr>
<td>Three Structures in One Diagram</td>
<td><a href="http://www.ucop.edu/~wgayle/allmaps.html">http://www.ucop.edu/~wgayle/allmaps.html</a></td>
</tr>
</tbody>
</table>

In each of the above-listed diagrams the boxes representing the various tables serve as links to lists of the table content, specified by DB2 column name and showing data type and cross-reference (to TCS from TCT and to TCT from TCS).

Description of Changes

The main changes to the TCT structure can be summarized as follows:

- The "common title information" is being stored in more functional manner, with the update timestamp eliminated as a part of the key. The Common Information Effective Date has been added to the key so that the table will continue to serve an historical function.

- For non-academic titles a level of structure has been added to store those elements which are qualified by Sub Location and Pay Representation Code and Effective Date.

- Non-academic title data are not segregated by Personnel Program Code but rather by whether the title is step-based or grade-based with respect to rates.

A detailed verbal discussion of the changes to each TCT table follows.

**PPPTCA**

The PPPTCA table will be eliminated.

**PPPTCI**

With the elimination of the PPPTCA table, PPPTCI becomes the root table of the TCT. For each title code there will be at least one row in PPPTCI. In addition to this change in role, a number of content changes are being made to PPPTCI.
Several columns on the PPPTCI table will be made obsolete or moved to other tables, some new columns will be added and the use of one column will be altered. The changes are detailed in the following sub-sections.

**Changes to the Primary Key**

The primary key of the PPPTCI table will be the column TCI_TITLE_CODE. In the current table the key is a combination of the TCI_TITLE_CODE and the TCI_UPDT_TIMESTAMP. This structure effectively makes this table a history of common information related to titles and needlessly complicates table access. TCI_UPDT_TIMESTAMP will not be made obsolete. It will continue to be populated with the appropriate value when a PPPTCI row is updated or inserted. However, because it will not be part of the primary key, access to common title information will be made easier and more efficient.

**Obsolete Columns**

The following paragraphs detail the PPPTCI columns which are to be made obsolete. A brief rationale for each item made obsolete is provided.

The Fair Labor Standards Act Status Code, TCI_FLSA_STATUS_CD, is used by PPS to determine whether an employee is eligible to receive overtime payments. In the latest version of TCS the column from which this column is derived, CSB_OVRTM_EXMPT, is qualified by Sub Location (a new element) and Pay Representation Code. Therefore, in TCT this column is removed from the PPPTCI table and placed on the new PPPTSL table, which includes Sub Location and Pay Representation Code as part of its primary key.

The Abolished Title Flag, TCI_TITLE_ABOL_FLG, is a superfluous element in the TCT. It is paired with the Abolished Title Date, TCI_TITLE_ABOL_DT, and conveys no information that cannot be determined from the presence of a non-initial value in that latter element.

The case of the Frozen Title Flag, TCI_FRZN_TITLE_FLG, is exactly analogous to that of the Abolished Title Flag.

UC Location Code, TCI_UC_LOCATION, seems to contain either the numeric campus location (i.e., '01', '02', '03', etc.) or spaces. It is not used by PPS and doesn't add any perceptible value.

As in the case of the FLSA Status Code, the Overtime Exemption Code, TCI_OVRTM_EXMPT_CD is being removed from the PPPTCI table and placed on the new PPPTSL table because its TCS counterpart is qualified by Sub Location and Pay Representation Code.

The Payscale Code, TCI_PAYSCALE_CODE, has value within TCS because it serves to identify the manner in which rates are stored for a particular title. However, since TCT will begin storing rates in all useful formats (i.e., monthly, hourly and either biweekly or semimonthly, depending on location) it will not be needed in TCT.

**Newly Added Columns**

The following columns are added:
The Supervisor Flag, **TCI_SUPERVISOR_FLAG**, is being moved to the PPPTCI table from the defunct PPPTPS table to reflect the fact that this information is treated as common to all titles within TCS.

The Health Flag, **TCI_HEALTH_FLAG**, which is currently stored as the A&PS Health Flag on the soon to be defunct PPPTPS table, is moved to the common information level.

A Update Source column is being added to each TCT table to capture a character string which will identify the program (or eventually, the userid) which caused the update or insertion of the row in question. On the PPPTCI table this column will be named **TCI_UPDT_SOURCE**.

A column is being added to the TCI table to indicate, titles which carry a Title Unit Code of '99', the non-99 bargaining unit, if any, to which the title is related. This column will be named **TCI_RELATED_UNIT**.

**TCT Tables Dropped**

The following tables in the current TCT are being eliminated:

- PPPTCA - with no replacement; PPPTCI will become the top table in the TCT hierarchy
- PPPTPM - with no replacement
- PPPTPS - with most elements moving to the new table PPPTSB, some to PPPTCI and PPPTSL
- PPPTPT - with elements moving to the new table PPPTSB
- PPPTPU - with elements moving to the new table PPPTRR
- PPPTPX - with elements moving to the new table PPPTSR
- PPPTPY - with elements moving to the new table PPPTOR

**TCT Tables Added**

The following tables are new to the TCT structure:

- PPPTGB will identify each grade-based non-academic title and indicate the Grades and Grade Types which pertain to the title.
- PPPTOR will carry all Time On Call rates for non-academic titles.
- PPPTRR will carry all regular pay rates for step-based non-academic titles.
- PPPTSB will identify all step-based non-academic titles and define the regular pay rate sets for those titles.
- PPPTSL will categorize all non-academic titles by Sub Location and Pay Representation Code.
• PPPTSR will carry all shift differential rates for non-academic titles.
Program Changes

New and modified programs in release 1304 will be discussed in the following categories

- One-time program to convert TCT data,
- Programs which update the TCT,
- PPTCTUTL, the TCT access utility,
- Programs converted from direct TCT access to using the TCT utility for access, and
- Programs made obsolete in this release.

One-Time Program for TCT Conversion

Program PPOT1304 is designed to read the old TCT and to write datasets which will serve as load files for the new TCT structure.

TCT Update Process

The TCT update process has been radically modified. There are two major reasons for the changes to this process:

1. changes to the TCT structure, the addition of new elements and the renaming of several tables, and
2. a desire to have the TCT update process conform to the requirements of the new Control Table update process, which will be released in the coming months.

The structural changes to TCT have already been discussed. The impact of these changes on the update process is most visible in the revisions to the transactions and the UPAY forms which describe them. The form revisions are discussed in detail below.

The impact of the new CTL update architecture is also extensive and described in a section immediately following the one on forms and transactions.

Forms/Transaction Changes

Several of the transactions which are used to update the TCT will be changed to accommodate the presence of new data elements and to respond to changes in key structure and table structure. The following table lists the changing forms and provides links to record layouts for the revised transactions:

<table>
<thead>
<tr>
<th>Form Number</th>
<th>Form Name (Revised)</th>
<th>URL for New Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPAY78 7</td>
<td>TITLE CODE TABLE, ALL TITLES - COMMON INFORMATION</td>
<td><a href="http://www.ucop.edu/payroll/forms/upay787.pdf">http://www.ucop.edu/payroll/forms/upay787.pdf</a></td>
</tr>
</tbody>
</table>
Update Programs and Components

All of the programs described in this section are new to PPS. The program name PPP004 has been re-used but the version of PPP004 issued with release 1304 contains none of the code from the previous version.

The new CTL update architecture calls for a main driver program, PPP010, and a set of generic called programs to deal with those processes which are common to all Control Table updates. Meanwhile, a set of table-specific called modules will handle the specifics of each logical Control Table. This architecture is respected in the new TCT update process, with a new version of PPP004 acting as a temporary stand-in for PPP010.

Generic Control Table Update Programs

The following paragraphs describe that portion of the TCT update process which mimics the future PPP010 process and is not specific to TCT but rather generic to all Control Table update processes.

PPP004

This program is a clone of the soon-to-be-released version of PPP010. It's functions are as follows:

- Determine the list of tables which might be updated

In the new version of PPP010 this will be accomplished by reading a not-yet-released DB2 table which will be called the Table of Tables, PPPTOT. Because PPP004 deals only with the Title Code and Salary Grade Tables, the data which will eventually be taken from the Table of Tables is stored in PPP004 working storage.

The Table of Tables (TOT) is keyed by the logical table number for the Control Table, e.g., '01' for the System Parameters Table, '02' for the Gross To Net Table, and so on. The TOT will carry the following information for each logical Control Table:

- The table name
- A flag which may indicate that the table is 'inactive', i.e., not subject to updates
- Ten report codes, which can be used to define various formats for listings of the table
• Sort control information, which defines the transaction fields to be used in sorting incoming transactions and the order in which they are to be used. In this area it is also possible to specify an override to the transaction sort. This override is intended to trigger the execution of table-specific code in the PPCTLSRT program.

• Consistency edit control information which is not used in by either the Title Code Table or Salary Grade Table update processes.

• Sort the incoming CTL update transactions

PPP004 calls PPCTLSRT to accomplish this task. See below for a detailed description of PPCTLSRT. PPCTLSRT applies sort controls appropriate to each logical table as stored in the Table of Tables.

• Process table updates

For any given logical Control Table, PPP004 will call a transaction handler program which reads in the sorted transactions for that table and manages the process of editing and updating. The transaction handler program name is made up of the string PPCTT plus the table number, e.g., '33' for the Title Code Table, '34' for the Salary Grade Table.

PPP004 initiates the reading of the sorted transaction file with a call to PPCTLTRD. Control is then passed to the appropriate transaction handler program for the table in question. PPCTLTRD is called by the transaction handler until there are no more transactions for that logical Control Table.

• Print table listings

For those control tables which have been updated, assuming that the user does not suppress the listing via run specification option, PPP004 calls a report listing program to list the table. Each logical Control Table has its own list program. Listing program names are made from the string 'PPCTR' and the logical table number. PPCTR33 is the report listing program for the Title Code Table; PPCTR34 fills that role for the Salary Grade Table.

• Print update edit errors and report statistics

Program PPCTRLRPT is called by PPP004 and by subordinate programs to generate a listing of updates accomplished, edit errors and table-by-table and overall statistics.

• Write mass change transactions as required by various Control Table updates

Program PPCTLMTW is called by individual Control Table transaction handlers to write mass change transactions as necessary and finally called by PPP004 to close the mass change transaction output file.

PPCTLMTW

Program PPCTLMTW is called primarily by the transaction handler programs to write mass
change transactions when required. The linkage area for this program is defined by the copy statement CPLNCMTW. Three call types are allowed for PPCTLMTW:

1. Initialization call
   On this call the program opens the Mass Change Transaction file, initializes transaction counters and writes a record to the Mass Change Transaction file.

2. Transaction call
   On this call the program writes a record to the Mass Change Transaction file.

3. Final call
   On this call the program closes the Mass Change Transaction file and returns the total count of transactions written to the caller.

**PPCTRLRPT**

Program PPCTRLRPT is used by all modules in the TCT update process to display transaction images, issue error messages, display transaction update statistics, etc. PPCTRLRPT produces a PPP0100 report which displays all such information.

The linkage to PPCTRLRPT is defined in the CPLNCTRI copy statement. Four call types are allowed for this program:

1. Initialization call
   This call is made only by PPP004. When called in this fashion PPCTRLRPT opens the file associated with the WRNPRT DD and headers are initialized.

2. Print call
   This call may be made by any module in the overall set which are involved in Control Table updating. Three actions are possible with this call:
   
   1. If the transaction field of the CPLNCTRI interface is non-blank, then the contents of that area are printed.
   
   2. If the message field of the CPLNCTRI interface is non-blank, then the message utility is called and, if the severity of the message if not 'Suppress Print', the message is displayed and the severity is returned to the caller.
   
   3. If the text field of the CPLNCTRI interface is non-blank, then the content of that field is printed.

3. Break call
   This call causes the module to begin a new page. If the text field of the CPLNCTRI interface is non-blank, the contents replace the contents of the fourth header line, are printed on the new page, and retained to be printed on every following page, until
changed again. NOTE: If the text field contains a single asterisk (*) in the first position, the contents of the fourth header line is deleted.

The input transaction, transaction disposition and message fields are ignored on this call.

The severity field always returns a zero (0).

4. Final call mode

The F (Final) call must be the last call to the print module. If the text field of the CPLNCTRI interface is non-blank, the contents is printed as the final line of the report. The print file is closed and all other termination tasks are done. The input transaction, transaction Disposition and message fields are ignored on this call. The highest message severity encountered by the program during any previous print calls is returned in the severity field.

PPCTLSRT

Program PPCTLSRT is called by PPP004 to sort incoming Control Table update transactions. The linkage between PPP004 and PPCTLSRT is defined in the CPLNCTSI copy statement. In addition, PPCTLSRT makes use of data from the Table of Tables which is placed in an external area by PPP004. That external area is defined by the copy statement CPWSTOFT.

PPCTLSRT opens the incoming transaction dataset and reads that file until it is empty. For each transaction read, PPCTLSRT uses Table of Tables data to develop a sort key and then releases a record to sort. The sorted record is built from the sort key and the original transaction image. The SORT step yields the sorted transaction file which has a logical record length of 114 bytes.

PPCTLSRT calls PPCTLRPT to print messages as needed.

PPCTLTRD

Program PPCTLTRD is a generic transaction reading module which is called by PPP004 and by all Control Table transaction handler programs to read the sorted transaction file produced by PPCTLSRT. The linkage to PPCTLTRD is defined by the new copy statement CPLNCSRI.

1. Initialization mode

This call is made only by PPP004. When called in this fashion PPCTLTRD opens the sorted transaction file, initializes a transaction counter, reads a single transaction record and returns control to the called.

2. Transaction read mode

This call should be made by the various table transaction handlers. PPCTLTRD will read a single transaction and add one to the transaction counter.

3. Final call mode

This call is made only by PPP004. If the sorted transaction file is empty (discovered in an earlier transaction read call) the linkage transaction counter will be updated from the
internal PPCTLTRD transaction counter.

Programs Specific to Particular Control Tables

The programs described below perform functions specific to either the Title Code Table or the Salary Grade Table. The programs fall into four functional categories:

1. Transaction Handlers
2. Editors
3. Updaters
4. Print processors

Transaction Handler Programs

PPCTT33

Program PPCTT33 is the transaction handler for the Title Code Table. PPCTT33 is called by PPP004 when transactions for table 33 are encountered in the mix of CTL update transactions.

PPCTT33 calls PPCTLTRD (see description above) to return sorted transactions. For each transaction, PPCTT33 calls PPCTRLRPT to display the transaction image. Then, the program determines whether the transaction is legitimate by examining the action code and the transaction type. If the transaction is not legitimate, a message is displayed and the next transaction is returned. If the transaction is legitimate, it is reviewed to determine whether it starts a set or continues a set and, in the latter case, whether it is the expected continuation of the set in process. If the transaction is accepted as part of the set, the data from the transaction is moved to an external work area which is shared with the appropriate edit module. If the transaction set passes editing the appropriate update module is called. For each complete set of transactions, the PPCTT33 calls the appropriate edit program.

There is one edit and one update module for each physical DB2 table which is a part of the logical Control Table. The edit and update programs for the Title Code Table are listed in the table below, together with the physical DB2 table to which they relate and name of the copy statement which defines the transaction input external work area which is loaded by PPCTT33.

<table>
<thead>
<tr>
<th>DB2 Table</th>
<th>Input Data Copy Member</th>
<th>Edit Module</th>
<th>Update Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPPTCI</td>
<td>CPCTTCII</td>
<td>PPCTTCIE</td>
<td>PPCTTCIU</td>
</tr>
<tr>
<td>PPPTGB</td>
<td>CPCTTGBI</td>
<td>PPCTTGBE</td>
<td>PPCTTGBU</td>
</tr>
<tr>
<td>PPPTOR</td>
<td>CPCTTORI</td>
<td>PPCTTORE</td>
<td>PPCTTORU</td>
</tr>
<tr>
<td>PPPTPA</td>
<td>CPCTTPAI</td>
<td>PPCTTPAE</td>
<td>PPCTTPAU</td>
</tr>
<tr>
<td>PPPTPB</td>
<td>CPCTTPBI</td>
<td>PPCTTPBE</td>
<td>PPCTTPBU</td>
</tr>
</tbody>
</table>
Editors

Each of the edit modules described below is built on the same pattern. Each module focuses on a single physical DB2 table. Data for that table is placed in an external work area by the transaction handler. There is a linkage interface between the transaction handler and the editor which is common to all tables. This linkage area, which is described in copy statement CPLNKCTE, contains such information as call type, return code, error message numbers, etc. If the editing is successful the edit module will set the 88 level value KCTE-STATUS-COMPLETE to TRUE. Otherwise, KCTE-STATUS-NOT-COMPLETED will be set to TRUE. The edit module moves data from the external input work area to another external area which defines the DB2 table for which the module is editing.

Update Programs

Each of the update programs is called by the transaction handler using the linkage area defined in copy statement CPLNKCTU. This copy statement contains such information as the call type, error messages issued, return code, etc. The data to be updated is contained in an external area defined by the INCLUDE member for the physical DB2 table to be updated.

Print Processors

For each logical Control Table (e.g., the Title Code Table as a whole) there is a single print processor which executes when requested or when the related Control Table has been updated and the print has not been suppressed. The print process for the Title Code Table is PPCTR33. The print processor for the Salary Grade Table is PPCTR34.

Changes to the Title Code Table Utility, PPTCTUTL

The flexibility of the Title Code Table utility program, PPTCTUTL, has been enhanced with the aim of improving performance of PPS processes and providing new functionality. Also, the external communication area used by PPTCTUTL and its callers has been modified to accommodate changes to the structure of the Title Code Table and to support several new call types.
**New Call Types**

The following new call types have been added to PPTCTUTL:

**Read Chosen Medium**

Returns common information for the Title Code (PPPTCI table content), Sub Location specific information (PPPTSL table content) for non-academic titles and Academic common information (PPPTPA table content) for academic titles.

**Frame Regular Rates**

Returns same information as 'Read Chosen Medium' plus the rate range or rate set for those titles for which regular rates are stored in TCT. That is, for step-based titles, whether academic or non-academic, all rates in the appropriate rate set are returned. For grade-based titles the minimum, mid-point and maximum rates of the appropriate range are returned.

**Read Shift Rate**

Returns same information as 'Read Chosen Medium' plus the shift differential rate for the title code, assuming that a valid key including Shift Type Code ('DA', 'EV', 'NT', 'SP', 'WD', 'WE' or 'WN') is supplied.

**Read On Call Rate**

Returns same information as 'Read Chosen Medium' plus the on call rate rate for the title code, assuming that a valid key including On Call Code ('1', '2' or '3') is supplied.

Although the "long" call types are still available in PPTCTUTL and the CPLNKTC copy statement still contains data definitions necessary to support these calls, as of release 1304 no PPS program uses them. In general, these calls are inefficient and their use in local programming is not advised.

**Programs to be Converted to Use PPTCTUTL**

A number of programs in PPS were found to access one or more of the TCT tables using static SQL statements, rather than using the TCT utility program, PPTCTUTL. For the most part these programs were obtaining data from the PPPTCI table. In some cases other tables were being accessed as well. As part of release 1304, such direct access is eliminated in all cases outside of the TCT update process. Calls to PPTCTUTL are substituted.

The modifications to the following programs were more or less identical. The changes are described in brief for each affected program:

**D21H04E**

The declaration of TCI-CURS, the cursor defined against the TCI table, was eliminated as was the INCLUDE statement for the TCI table. A working storage variable was added to store the name of the TCT utility program. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL.
In the 8100-OPEN-CURS paragraph, static SQL selecting current date from the view PPPVZTCI_TCI and opening the cursor TCI-CURS was commented out and a statement added setting the PPTCTUTL call type to XTCL-READ-ALL-TITLES-FIRST.

In the 8200-FETCH-CURS paragraph, static SQL fetching a row from TCI-CURS was commented out and a call to PPTCTUTL was substituted. In addition, an IF statement was added following the PPTCTUTL call to interrogate the XTCL-EVENT-CODE value. If the value is found to be XTCL-READ-ALL-TITLES-FIRST, it is changed to XTCL-READ-ALL-TITLES-NEXT. The SQL code returned from PPTCTUTL is moved to SQLCODE.

In the 8300-BUILD-TSQ-ROW paragraph the STRING statement involving TCI table data is commented out and a STRING statement using data from the PPTCTUTL interface is substituted.

Finally, in the 8400-CLOSE-CURS paragraph, the static SQL statement closing the TCI-CURS cursor is commented out.

**PPDXAPPT**

The INCLUDE statements for the TCI table was commented out. A working storage variable was added to store the name of the TCT utility program. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL.

In the 2400-PROCESS-APPT section, code moving an appointment Title Code to the host variable TCI-TITLE-CODE was commented out. A move of the Title Code to the PPTCTUTL interface variable XTCL-CHOSEN-TITLE-CODE was substituted.

In the 6200-SELECT-TCI section is replaced by a section named 6200-GET-TITLE-DESCRIPTION which executes a call to PPTCTUTL. This section is performed from 2400-PROCESS-APPT. Upon a successful PPTCTUTL call the content of XTCL-TITLE-NAME-ABBREVIATED is moved to HR-LINEA-DESCR1.

**PPDXHIRE**

The INCLUDE statement for the TCI was commented out. A working storage variable was added to store the name of the TCT utility program. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL.

In the 2400-PROCESS-APPT section, code moving an appointment Title Code to the host variable TCI-TITLE-CODE was commented out. A move of the Title Code to the PPTCTUTL interface variable XTCL-CHOSEN-TITLE-CODE was substituted.

In the 6200-SELECT-TCI section is replaced by a section named 6200-GET-TITLE-DESCRIPTION which executes a call to PPTCTUTL. This section is performed from 2400-PROCESS-APPT. Upon a successful PPTCTUTL call the content of XTCL-TITLE-NAME-ABBREVIATED is moved to HR-LINEA-DESCR1.

**PPPGSR01**

The INCLUDE statements for the TCI were commented out. A working storage variable was
added to store the name of the TCT utility program. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL.

In the 2410-PROCESS-CURRENT-APPT section, code moving an appointment Title Code to the host variable TCI-TITLE-CODE was commented out. A move of the Title Code to the PPTCTUTL interface variable XTCL-CHOSEN-TITLE-CODE was substituted.

In the 6200-SELECT-TCI section is replaced by a section named 6200-GET-TITLE-DESCRIPTION which executes a call to PPTCTUTL. This section is performed from 2410-PROCESS-CURRENT-APPT. Upon a successful PPTCTUTL call the content of XTCL-TITLE-NAME-ABBREVIATED is moved to GSR1-APPT-TITLE-DESC.

**PPPVREDO**

The INCLUDE statements for the TCI were commented out. A working storage variable was added to store the name of the TCT utility program. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL.

In the 4020-TITLE-CODE-EDIT section, code moving an appointment Title Code to the host variable TCI-TITLE-CODE was commented out. A move of the Title Code to the PPTCTUTL interface variable XTCL-CHOSEN-TITLE-CODE was substituted.

Section 9020-SELECT-TCI is eliminated. A call to PPTCTUTL is made in 4020-TITLE-CODE-EDIT. If the Title Code is not found (XTCL-TITLE-NOT-FOUND is TRUE), CPWSDEED-ELEM-EDIT-ERROR is set to TRUE for the appropriate occurrence of the CPWSDEED element array.

**PPP031**

The INCLUDE statements for the TCI were commented out. A working storage variable was added to store the name of the TCT utility program. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. The TCI-CURSOR-ROWS-SW variable and its 88 level definitions were commented out as was the declaration of TCI_CURSOR.

Section 20000-PROCESS-TCI-CURSOR has been renamed 20000-PROCESS-TITLES. All procedure division code involving direct SQL access to the PPPTCI table was commented out. Code calling PPTCTUTL with a call type of XTCL-READ-ALL-TITLES-FIRST has replaced the OPEN TCI_CURSOR statement. All subsequent calls to PPTCTUTL are made in the XTCL-READ-ALL-TITLES-NEXT mode.

References to fields defined in the PPPVZTCI include statement have been commented out and replaced with references to the corresponding fields in the CPLNKTCI copy statement.

Procedure division code changes in this program are confined to the 10000-INITIALIZATION and the 20000-PROCESS-TITLES sections.

**PPP864**

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL.
In the section 1875-UPDATE-TEACHING-ASSISTANT, the variable XPAR-TTL-CD (appropriately subscripted) is moved to XTCL-CHOSEN-TITLE-CODE rather than to TCI-TITLE-CODE. The perform of section 9700-SELECT-TCI is commented out as is that section itself. A call to PPTCTUTL is made instead. The field XTCL-CTO-OSC is used to determine whether the Title Code qualifies as a Graduate Student title.

PPRCADVC

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 9600-SELECT-TCI and all references to it were commented out. In section 5240-EARNINGS-DTL-ROW a call to PPTCTUTL was added to load the TCI-TITLE-NM-ABBRV value into XTCL-TITLE-NAME-ABBREVIATED.

PPWEAPC

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 9000-SELECT-TCI and all references to it were commented out. In section 1520-TITLE-CODE-TRANSLATION a call to PPTCTUTL was added to load the TCI-TITLE-NM-ABBRV value into XTCL-TITLE-NAME-ABBREVIATED.

Because the PPPTCI table access was the only procedure division SQL in this program, the includes of CPPDXE99 and CPPDXP99 have been commented. These copy statements contain the procedure division statements for handling negative SQLCODE conditions. This program still requires a DB2 pre-compile because of data division include statements. However, it no longer requires a package bind.

PPWEAPP

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 9000-SELECT-TCI and all references to it were commented out. In section 1520-TITLE-CODE-TRANSLATION a call to PPTCTUTL was added to load the TCI-TITLE-NM-ABBRV value into XTCL-TITLE-NAME-ABBREVIATED.

Because the PPPTCI table access was the only procedure division SQL in this program, the includes of CPPDXE99 and CPPDXP99 have been commented. These copy statements contain the procedure division statements for handling negative SQLCODE conditions. This program still requires a DB2 pre-compile because of data division include statements. However, it no longer requires a package bind.

PPWELOF
The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

In section 1511-TITLE-CODE-TRANSLATION a call to PPTCTUTL was added to load the TCI-TITLE-NM-ABBRV value into XTCL-TITLE-NAME-ABBREVIATED. Static SQL for accessing the PPPTCI table was commented out.

**PPWEPER**

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8200-SELECT-TCI and all references to it were commented out. In section 1600-UPDATE-TRANSLATIONS a call to PPTCTUTL was added to load the TCI-TITLE-NM-ABBRV value into XTCL-TITLE-NAME-ABBREVIATED.

**PPWHADC**

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the a new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

**PPWHADF**

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the a new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

**PPWHLOA**

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the a new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

**PPWHPER**

The INCLUDE statements for the TCI were commented out. A COPY statement was added to
define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

PPWHSEP

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

PPWHSUM

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

PPWIAPP

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

PPWIAPS

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was added to store the name of the Title Code Table utility program.

The section 8800-SELECT-TCI and all references to it were commented out. It was replaced by the new section called 8800-GET-TITLE-DESCRIPTION, in which a call is made to PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.

PPWIAPT

The INCLUDE statements for the TCI were commented out. A COPY statement was added to define the external area which serves as linkage to PPTCTUTL. A working storage variable was
added to store the name of the Title Code Table utility program.

In the section 8250-GET-TTL-DATA the static SQL which accessed the PPPTCI table was commented out and replaced with instructions calling PPTCTUTL to populate the field XTCL-TITLE-NAME-ABBREVIATED.