PPPIVR
Design Document

Document IVRDD1
August 18, 1995

FINAL

Information Systems & Computing
Office of the President
University of California
Introduction

In early 1995, UC Benefits began a set of parallel projects to make IVR (Interactive Voice Response) services available to UC employees and annuitants. One project was to develop an IVR system to provide employees and annuitants with access to information relating to their retirement accounts and other information residing within the UCRP retirement system. The second project was to develop an IVR system which would allow employees to access information related to their current benefits and would allow the annual Open Enrollment process to be completed via IVR rather than the traditional forms-based process.

This second project resulted in the creation of bencom, an IVR system which will allow employees to obtain general benefits information, campus-specific benefits information, and information about their current enrollment and participation in all benefits programs for which they are eligible. The system will also allow employees to complete Open Enrollment in November, 1995.

Bencom is not a single IVR system, but rather a network of six such systems. There are servers located at UCOP, Berkeley, Los Angeles, Irvine, Riverside, and San Diego. The UCOP installation serves those campuses currently running their payroll systems on the UCOP hardware: Davis, San Francisco, Santa Barbara, and Santa Cruz, as well as Hastings and ASUCLA. It is anticipated that IVR support for the DOE Laboratories (Berkeley, Livermore, and Los Alamos) will be added in 1996.

The bencom servers are PC-based, running OS/2 and Edify IVR software. Applications programming, using Edify IVR programming tools, is being done by a team of users, programmers, and contractors at UCOP. Each of the six bencom servers will run the same basic application, although there will be campus-specific recorded information available to the employees at each site.

This document discusses the programming and operational support necessary to provide individual, employee-level information to bencom callers. The source of this data is the Base Payroll/Personnel System (PPS).
Data Model

During the design of bencom, a variety of data models were considered. The most desirable model would be one which allows the data residing on the mainframes, which run PPS, to remain there and be retrieved "as needed." Since the Edify software supports data retrieval via the technique commonly referred to as "screen scraping," this was the first option considered, since the PPS system is an on-line, interactive system, and all of the necessary data could be made available via one or two specifically designed CICS screens. The major drawback to this technique is that it only works when CICS is available, which at most campuses, is a daytime window, for example 7 a.m. until 6 p.m. Since one of the reasons for providing an IVR is to allow as close to 24-hour access as possible, having the only means of data access be via a CICS screen was obviously inadequate.

The IVR servers all have DB2/2 available and the PPS data on the mainframe is stored in DB/2, so the next possible consideration was some technique which would allow the IVR servers to "directly" access the DB/2 data on the mainframe. Such a technology is available from IBM in the form of the DDCS/2 and DRDA products, which do allow such direct access. Unfortunately, these products currently require LU 6.2 connectivity, which many campuses are moving away from, in favor of TCP/IP networks. There has been some indication from IBM that a TCP/IP version of DRDA may be available in the future. If and when such a product does become available, this option should be reconsidered.

Given that no practical method of direct access to the mainframe data is available to the IVR servers, it is obvious that the necessary subset of data needs to be made available via some other method. Two such methods were considered.

Since each of the IVR servers has DB2/2 available, the first method considered was to "download" an extract of the required PPS data from the DB/2 database on the mainframe to the DB2/2 database on the IVR server. The advantage to this method is that the data for a campus with an IVR server remains on that campus' server. A potential disadvantage is that campus data processing personnel have to become much more intimately involved with the IVR server than would be hoped. Despite that disadvantage, this method became the candidate of choice until the University was advised by Edify technical consultants that there were some very significant limitations to table size using DB2/2 databases. There is no absolute limit, but DB2/2 seems to break down when tables exceed about 40,000 rows. Examining the table size projections which had been made early in the process, it was obvious that smaller campuses, such as Riverside, would have no problem, medium-sized campuses such as Davis would be "on the edge," and the two largest campuses, Berkeley and Los Angeles, would be definitely over the limit. The IVR server at UCOP posed a particular problem because the database there needed to store data for four campuses (Davis, San Francisco, Santa Barbara, and Santa Cruz), in addition to Hastings and ASUCLA. While none of those sites by themselves were projected to exceed the 40,000 record limit, taken as a group, the limit was definitely going to be exceeded. The alternative here was to create separate "databases" for each site, so that individual tables would remain under the limit, but this approach would have meant that the Edify application program would be
different at UCOP than at the separate campus sites, something which was not considered desirable.

At about this same time, the PPS Student FICA project was being completed and this provided additional impetus to select an alternative data storage process, because it meant that every student who made a DCP contribution was now a candidate to be included in the IVR database.

The remaining option which was available to the team was to plan to store all of the employee data on a centralized Sybase server, which the IVR could access. Successful access to data on the UCOP Sybase server from the campus IVR systems had been demonstrated by the earlier UCRP Board Election application which ran on the bencom servers.

Centralizing the data provided advantages in terms of data backup and integrity, as well as reducing the need for campus data processing staff to interact with the IVR on a frequent basis. This was the option chosen, although again it should be pointed out that if an appropriate DRDA product does become available, the direct approach to the data would be preferable and should be considered.

**Database Design**

Once the decision was made to locate all of the data on the UCOP Sybase server, the design of that database was undertaken. In order to keep the IVR application programming as simple as possible, it was decided that there would be a single database on Sybase, containing the data for all locations. It should be noted here that although the data for all locations is being coalesced into a single database, each record in the database is identified by a location code, and the IVR server at a given location will only have access to data for employees at that location. Identification of location may be indirect, as is the case at Irvine, Los Angeles, Riverside, and San Diego, where the IVR server will assume the appropriate location code, or direct, at the other sites. Employees calling the Berkeley IVR will be asked to identify whether they are Berkeley Campus or UCOP employees. This is for purpose of deciding which “local messages” to play for the employee; for purposes of data access, all employees are assumed to be from Berkeley. Employees calling the UCOP IVR server will be asked to identify the campus at which they are employed. This identification will be used both for determining which “local messages” to play and for data access.

The single database will contain three tables of employee-related data extracted from PPS: the Employee table, the Dependent table, and the Deduction table.

**Employee Table**

The Employee table contains one row per employee. The table key is location code and employee Social Security Number. Attachment 1 is a listing of the SQL used to create this table on Sybase.
Dependent Table

The Dependent table contains one row for each dependent for each employee. The table key is location code, employee Social Security Number and dependent number. Attachment 2 is a listing of the SQL used to create this table on Sybase.

Deduction Table

The Deduction table contains one row for each benefits-related deduction for each employee. The table key is location code, employee Social Security Number, “group code,” and gross-to-net number. “Group code” is an artificial division of benefits-related deductions into presentation groups for use by the IVR. Attachment 3 is a listing of the SQL used to create this table on Sybase. A later section of this document will identify the specific GTN entries chosen for the server and specify their groupings.

Other Data

In addition to the three tables which contain data extracted from PPS, this database contains several other tables, all of which are used by bencom in support of the InfoLine and Open Enrollment applications. These other tables fall into one of two categories: support tables and transaction tables.

Support tables are tables which contain data needed to support the IVR application. Examples of these tables include plan rates, carrier Primary Care Physician directories, carrier Zip Code tables, and other similar supporting information.

Transaction tables are tables which are specific to the Open Enrollment portion of bencom, and which are used to hold records of the choices made by employees during the open enrollment period.

Data Security

At this point, it is appropriate to discuss the various levels of security which are applied to the data stored on the UCOP Sybase server. First of all, the data used by the bencom IVR servers is stored in a database which is under the normal Sybase access controls. This database is "owned" by a database administrator, who is the only person outside the Sybase administrator having the ability to grant other access to the database. Such access is granted on an as-needed basis, to Sybase user IDs. As is normal, each such user ID is protected by a password. The IVR system is given one such user ID. That ID is granted select access to the three PPS data tables. Note that this is read-only access, the IVR has no ability to update PPS data.

As mentioned earlier, within the same database there are support tables, used to store other information needed by bencom. While this is not PPS data, the same type of access security is used. Access to this information is either given by grant of specific select authority or via the use of Sybase stored procedures. Once again, this data cannot be updated by the IVR.
Finally, also within this database reside tables used to hold records of the changes made by employees during the Open Enrollment period. The IVR, via its user ID, is granted update access on these tables so that employee choices can be recorded and updated during subsequent calls to the IVR system. Note that for purposes of backup, copies of these tables also reside on the IVR system's internal DB2/2 database.

**Campus Identification and PIN Usage**

In addition to the database access controls put in place through Sybase, there is a further level of security which is applied to control access to data for individual employees. Access to such individual data is controlled by requiring the employee to provide identification of the location code (either directly or indirectly), the employee Social Security Number, and a Personal Identification Number (PIN). In order to access individual employee data, all three pieces must match. There must be an employee at the given location code who has the provided Social Security Number, and, the provided PIN must be valid for the given Social Security Number.

It should be noted that PINs are not stored on the IVR’s PPS database. UCOP maintains a separate PIN server, which does happen to be a Sybase database, but which is “owned” by a different administrator. The common PIN server is used for both the UCRP IVR and bencom, and is available for other uses as well. Initial PINs were assigned to all employees with UCRP records and annuitants in late July, 1995, and the initial PIN database was loaded at that time. Newly-hired or newly-eligible employees are assigned an initial PIN of 0000 and are required to select a new PIN value the first time they call either of the IVR systems. An employee can perform a PIN change during any call to either IVR system, as long as the correct value of the current PIN can be provided.

For security purposes, the IVR systems are not granted select access to the PIN database. All access is via Sybase stored procedures which allow only very specific forms of access. Since calls to these stored procedures are sent from remote sites to the PIN server over the internet, the PIN value is encrypted prior to transmission, as a means of providing additional security.

**Data Backup**

All data stored on the Sybase server is backed up as part of routine processes in place at the UCOP Data Center. Additionally, Sybase uses transaction logging to record all update activity to tables, thus providing an additional level of backup security. In terms of the PPS data, the ultimate backup is, of course, the campus PPS system, however UCOP will maintain updated master data files during the process of populating and maintaining the IVR database, which will serve as an immediate backup source for the PPS data.
Payroll System Modifications

The PPS data stored for use by bencom is a subset of regular PPS data along with some data which is derived from PPS data. The only PPS change is the addition of a single program which will extract and derive this data from existing PPS data and create the files of data which are used to populate and maintain the bencom database.

Program PPIIVR is modeled after the existing retirement interface program PPI740. The basic idea is that the initial run of the program produces files containing all employee data, to be sent to the IVR system, and “history” files, which record the data which has been sent to the IVR. Each subsequent run of the program then compares the current PPS data for the employee with that recorded in the history file, and an update record is only sent for an employee when there have been changes to that data. When an update record is sent, the history file is updated, to be ready for the next cycle. In the case of PPIIVR, there are three separate update files, one for each of the tables in the database, and three separate history files, corresponding to the three update files.

Figure 1 is a flowchart showing the “startup” run of PPIIVR, which creates update files which are used to initially load the bencom database, and the initial history files.

Figure 2 is a flowchart showing subsequent runs of PPIIVR, using the input history files created by the previous run, and producing both update files to be used to update the bencom database, and updated history files.

In addition to the single program, PPIIVR, PPS changes include three copymembers, a bind member, and updates to the Payroll System Messages Table.

Copymembers

There is a separate copymember which describes the layout of each of the three files which are used to update the bencom database. These three copymembers are

- CPWSIVRE the Employee file (Attachment 4)
- CPWSIVRD the Dependent file (Attachment 5)
- CPWSIVRG the Deduction file (Attachment 6)

Bind Member

Bind member PPIIVR is used to define the plan which must be bound for the program PPIIVR.

System Messages Table

PPIIVR can encounter a number of error conditions which can be the result of operations error, database failure, or inconsistent data found on the PPS databases. The majority of these error conditions are severe enough to cause the program to cease
execution, although a few are not. The messages which may be issued by the program and their explanation, are as follows:

**IV-001 Invalid Run Specification ID**
The provided run specification does not have the proper ID. The run is terminated with a “See Operations” level message.

**IV-002 Invalid Run Specification History Flag**
The history flag on the run specification must be either “Y”, indicating that input history files are to be processed, or “N”, indicating that no input history files are to be processed. The run is terminated with a “See Operations” level message.

**IV-003 Input Employee History File is Empty**
Input history files are being processed, but the provided employee history file is empty. The run is terminated with a “See Operations” level message.

**IV-004 Input Dependent History File is Empty**
Input history files are being processed, but the provided dependent history file is empty. The run is terminated with a “See Operations” level message.

**IV-005 Input Deduction History File is Empty**
Input history files are being processed, but the provided deduction history file is empty. The run is terminated with a “See Operations” level message.

**IV-006 History File Location Codes do not Match Campus**
The header record for at least one of the input history files does not match the campus location code obtained from the Campus Control Record table. The run is terminated with a “See Operations” level message.

**IV-007 Unable to Retrieve PER Table Data**
The program expected to find data on the PER table for the indicated employee ID number, but was unable to retrieve the data. This indicates a serious database problem. The run is terminated with a “See Systems” level message.

**IV-008 Current Date not Available from SCR**
The program requires the value of “SCR CURRENT DATE” from the SCR table, but was unable to retrieve the data. The run is terminated with a “See Systems” level message.

**IV-009 Location Code not Available from CCR**
The program requires the campus location code from the CCR table, but was unable to retrieve the data. The run is terminated with a “See Systems” level message.
IV-010 Unable to Open SSN Cursor
PPIIVR uses a cursor based on employee Social Security Number to process the EDB in SSN Order. The attempt to open that cursor failed. The run is terminated with a “See Systems” level message.

IV-011 PPBENE1PD has Failed
The benefits rate modules are used to calculate premium amounts and UC contributions for each benefit plan in which the employee is participating. Execution of the indicated module failed for the employee identified with the error message. The run is terminated with a “See Systems” level message.

IV-012 PPBENXDI has Failed
The benefits rate modules are used to calculate premium amounts and UC contributions for each benefit plan in which the employee is participating. Execution of the indicated module failed for the employee identified with the error message. The run is terminated with a “See Systems” level message.

IV-013 PPBENXLI has Failed
The benefits rate modules are used to calculate premium amounts and UC contributions for each benefit plan in which the employee is participating. Execution of the indicated module failed for the employee identified with the error message. The run is terminated with a “See Systems” level message.

IV-014 PPBENDLI has Failed
The benefits rate modules are used to calculate premium amounts and UC contributions for each benefit plan in which the employee is participating. Execution of the indicated module failed for the employee identified with the error message. The run is terminated with a “See Systems” level message.

IV-015 PPBENADD has Failed
The benefits rate modules are used to calculate premium amounts and UC contributions for each benefit plan in which the employee is participating. Execution of the indicated module failed for the employee identified with the error message. The run is terminated with a “See Systems” level message.

IV-016 PPBENXHI has Failed
The benefits rate modules are used to calculate premium amounts and UC contributions for each benefit plan in which the employee is participating. Execution of the indicated module failed for the employee identified with the error message. The run is terminated with a “See Systems” level message.
IV–017  PPBENVXVI has Failed
The benefits rate modules are used to calculate premium
amounts and UC contributions for each benefit plan in which
the employee is participating. Execution of the indicated module
failed for the employee identified with the error message. The
run is terminated with a “See Systems” level message.

IV–018  PPBENVXJD has Failed
The benefits rate modules are used to calculate premium
amounts and UC contributions for each benefit plan in which
the employee is participating. Execution of the indicated module
failed for the employee identified with the error message. The
run is terminated with a “See Systems” level message.

IV–019  Invalid Bond Number Encountered
If the employee is participating in the US Savings Bond program,
each bond being purchased must be extracted for the IVR. The
PPS system supports up to four bonds per employee, numbers
1–4. The indicated illegal bond number was found on the record
for the employee identified with the error message. The run is
terminated with a “See Systems” level message.

IV–020  Foreign Address Indicated but no FAD Data Found – Ignored
The indicated employee had a foreign address indicator value
which specified that foreign address data elements should be
present on the FAD table, but no such data was found. A
“Warning” level message is issued and the run continues. The
employee is reported as if the foreign address indicator were not
present.

IV–021  Unable to Translate Country Code to Country Name – Code
Used
The foreign country code given in the indicated employee’s
foreign address did not have a translation on the Foreign
Country Table (FCT). A “Warning” level message is issued and
the run continues. The foreign country code is used in the
address in place of the country name.

IV–022  PPDSAUTL has failed
PPIIVR uses PPDSAUTL to load information about all
deductions for the employee. The utility has failed for the
indicated employee. A “See Systems” level message is issued and
the run is terminated.
COBOL Programs

PPIIVR

PPIIVR is a single COBOL program driven by a run specification record which is used to identify the mode of the run being made. There are two possible modes: initial, and update. In initial mode, the program extracts complete records for all eligible employees on the EDB and creates initial history files. In update mode, data for eligible employees on the EDB is compared with data contained in the history files from the previous run, and update (add, change, and delete) records are generated where appropriate.

Employee Selection

PPIIVR, in its present configuration, selects a broad spectrum of employees for inclusion on the IVR database. It is possible that the selection of eligible employees will be narrowed in the future, once some experience with bencom is gained. PPIIVR selects all employees on the EDB who do not get filtered out by one of two screening tests: Social Security Number and data values.

The SSN screening is necessary because of the use of Social Security Number as the IVR system identifier. The PIN database was initially loaded from the UCRP system, which is keyed by Social Security Number and does not permit duplicates. The payroll system, on the other hand, is keyed by Employee ID Number which must be unique. There is nothing within PPS which prohibits more than one employee from being assigned the same Social Security Number on an individual campus, let alone anything which would prevent the assignment of the same Social Security Number to employees at two or more different campuses. Since the IVR data is keyed by location in addition to Social Security Number, no conflicts arise within the IVR database if two locations report the same Social Security Number (although the PIN database will be the final arbiter of which of the two employees gain access to the system). Problems would arise within the IVR database, on the other hand, if a single location were allowed to report two or more employees with the same Social Security Number. PPIIVR does not act as an arbiter of duplicate Social Security Numbers. If it finds two or more employees with the same Social Security Number, it simply rejects them all. In addition to duplicate Social Security Numbers, PPIIVR also rejects all employees where the Social Security Number is missing, which is defined as having a value of spaces or all zeroes.

Consideration was given to having PPIIVR produce a report of duplicate and missing Social Security Numbers, but it was felt that since PPIIVR runs on a daily basis in production, the report would be produced too frequently and would tend to be ignored. If sufficient interest is shown by campuses, a separate reporting program, probably a FOCUS report, will be developed to provide this information. Such a report could be run on an as-needed basis. In the absence of such a report, a couple of simple SPUFI queries can be used to identify missing and duplicate Social Security Numbers:

```sql
SELECT EMPLOYEE_ID FROM PPPPAY
WHERE SOC_SEC_NUMBER IN ('00000000', '00000000')
ORDER BY EMPLOYEE_ID;
```
will identify those employees where the Social Security Number is missing, while

```
SELECT EMPLOYEE_ID, SOC_SEC_NUM FROM PPPPAY PAY
WHERE SOC_SEC_NUMBER NOT IN ('00000000', '1') AND
1 <
(SELECT COUNT(*) FROM PPPPAY
WHERE SOC_SEC_NUM = PAY.SOC_SEC_NUM)
ORDER BY SOC_SEC_NUM, EMPLOYEE_ID;
```

will identify the duplicates.

Data element selection rejects all employees with an employment status of "I" (Inactive) or "S" (Separated), all employees with a name field consisting of all asterisks (these are the remnant records from ID number changes), and all employees with an invalid birthdate (because of the need for a valid employee age in order to calculate benefits premiums).

**Deduction Selection**

The IVR system is interested in a variety of different gross-to-net or deduction items. The list of desired items is quite diverse and there is no way that the gross-to-net number corresponding to the desired item can be identified through table attributes. Thus, selection must be done by gross-to-net number. To support the desired processing sequence on the IVR, these deduction items must be arranged in predefined groups. Furthermore, each group has its own requirement as to which deduction balances (G-, D-, or Y-) are wanted, and whether or not an effective date is present. The identified groupings, gross-to-net numbers, and desired balances are indicated in the following chart.

**Group 1: Flat-Dollar 403(b) Deductions—"G" and "Y" balance; effective date.**

- 001 Savings
- 002 Bond
- 003 Equity
- 228 Multi-Asset
- 246 Fidelity
- 292 Insurance Company Contract
- 293 Money Market
- 294 Calvert

**Group 2: Percentage-Based 403(b) Deductions—"G" and "Y" balance; effective date.**

- 229 Multi-Asset
- 232 Savings
- 233 Bond
- 234 Equity
- 235 Insurance Company Contract
- 236 Money Market
Group 3: DCP Retirement Deductions—"Y" balance; no effective date.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>DCP Career Savings</td>
</tr>
<tr>
<td>212</td>
<td>DCP Career Equity</td>
</tr>
<tr>
<td>213</td>
<td>DCP Career Bond</td>
</tr>
<tr>
<td>214</td>
<td>DCP Career Insurance Company Contract</td>
</tr>
<tr>
<td>215</td>
<td>DCP Career Money Market</td>
</tr>
<tr>
<td>216</td>
<td>DCP Career Multi-Asset</td>
</tr>
<tr>
<td>217</td>
<td>DCP Career Fidelity</td>
</tr>
<tr>
<td>218</td>
<td>DCP Career Calvert (Currently Unused)</td>
</tr>
<tr>
<td>036</td>
<td>DCP Casual Savings</td>
</tr>
<tr>
<td>037</td>
<td>DCP Casual Equity</td>
</tr>
<tr>
<td>038</td>
<td>DCP Casual Bond</td>
</tr>
<tr>
<td>039</td>
<td>DCP Casual Insurance Company Contract</td>
</tr>
<tr>
<td>155</td>
<td>DCP Casual Money Market</td>
</tr>
<tr>
<td>156</td>
<td>DCP Casual Multi-Asset</td>
</tr>
<tr>
<td>278</td>
<td>DCP Casual Fidelity</td>
</tr>
</tbody>
</table>

Group 4: Non Tax-Deferred Deductions—"G" and "Y" balance; effective date.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>023</td>
<td>Savings</td>
</tr>
<tr>
<td>030</td>
<td>Bond</td>
</tr>
<tr>
<td>031</td>
<td>Equity</td>
</tr>
<tr>
<td>202</td>
<td>Insurance Company Contract</td>
</tr>
<tr>
<td>203</td>
<td>Money Market</td>
</tr>
<tr>
<td>252</td>
<td>Multi-Asset</td>
</tr>
<tr>
<td>279</td>
<td>Fidelity</td>
</tr>
</tbody>
</table>

Group 5: Miscellaneous Group I—"G" and "Y" balance; effective date varies.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>284</td>
<td>Benham (has effective date)</td>
</tr>
<tr>
<td>239</td>
<td>IDS IRA (no effective date)</td>
</tr>
</tbody>
</table>

Group 6: UCRS Loans—"G", "D", and "Y" balance; no effective date.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>243</td>
<td>Short-Term Loan</td>
</tr>
<tr>
<td>253</td>
<td>Long-Term Loan</td>
</tr>
</tbody>
</table>

Group 7: Retirement Buybacks & Misc.—"G" and some "D" balances; no effective date.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>006</td>
<td>UCRS Retro (has &quot;D&quot; balance)</td>
</tr>
<tr>
<td>029</td>
<td>UCRS Buyback (has &quot;D&quot; balance)</td>
</tr>
<tr>
<td>026</td>
<td>PERS Survivor (no &quot;D&quot; balance)</td>
</tr>
<tr>
<td>033</td>
<td>PERS Additional (no &quot;D&quot; balance)</td>
</tr>
<tr>
<td>034</td>
<td>PERS Buyback (has &quot;D&quot; balance)</td>
</tr>
</tbody>
</table>
Group 8: Reserved for future use

Group 9: Miscellaneous Group II

008 403(b) Aggregate: "Y" balance is sum of the "Y" balances from groups 1 and 2 above. "D" balance is the "U" balance from the EDB—the maximum amount the employee is permitted to contribute during the calendar year (zero if no worksheet is filed). No effective date.

225 DepCare: "G" and "Y" balances, with effective date.

Although it is believed that most campuses use the "standard" gross-to-net numbers for these deductions, it is possible that some variation has been introduced due to local campus requirements. PPIIVR, therefore, will contain a table which links the group and the "base" GTN number to a campus GTN number. If a campus has "non-standard" GTN numbers for one or more of these deductions, it will be necessary for the campus to customize this table, by inserting their own local GTN numbers into the table, replacing the released "campus" GTN number. Note that bencom will only recognize the "base" GTN numbers.

Program Files

PPIIVR has four input datasets and creates seven output datasets:

The inputs are

- Run Specification Record (CARDFIL)
- Prior Employee History File (EMPHOLD)
- Prior Dependent History File (DEPHOLD)
- Prior Deduction History File (DEDHOLD)

The outputs are

- Errors and Run Controls Report (PRNTFIL)
- Employee History File (EMPHNEW)
- Dependent History File (DEPHNEW)
- Deduction History File (DEDHNEW)
- Employee Update File (EMPUPD)
- Dependent Update File (DEPUPD)
- Deduction Update File (DEDUPD)

Program Flow

PPIIVR is arranged in three major program sections:

- Initialization (10000-GENERAL-INITIALIZATION)
- Employee Processing (20000-PROCESS-EMPLOYEES)
- Termination (30000-GENERAL-TERMINATION)
Program Initialization

This portion of the program is concerned with the initialization of all program areas prior to the processing of employee records. Initialization steps are as follows:

1. Open the output report file and initialize the standard report headings

2. Read and validate the run specification record. If no run specification record is provided, the program assumes that input history files are to be processed. The run is terminated if the provided run specification is invalid or if the flag indicating whether or not input history files should be processed is invalid.

3. Retrieve the current processing month from the System Control Record and the campus location code from the Campus Control Record.

4. Initialize the files. This includes several steps:
   
   a. If input history files are being processed, open them, read the header records from each, check the header records for synchronization and terminate the run if the file set is not synchronized. Read the first data record from each file.

   The current processing month on the employee file history record is compared to that retrieved from the System Control Record, and if the processing month has changed (indicating that Monthly Periodic Maintenance has been run), a switch is set indicating that it is necessary to check all employees for updated data. If a new month has not been begun, then only employees showing that they have had an update activity processed (recognized by comparing the timestamp on the PER table with the prior timestamp, captured in the history file) or showing that they have had compute activity (recognized by comparing the last payroll activity date from the PCM table with the prior value, captured in the history file) will be checked for updated data.

   If no input history files are being processed, initialize the key fields for each input history file record to high values so that key comparison will cause all eligible employees to look like they should be “added” to the files.

   b. Open the output history files and write a header record to each.

   c. Open the output update files.

5. Open the Social Security Number cursor on the PAY table. This cursor will be the driving control for employee selection. The cursor selects Social Security Number and Employee ID Number for all employees where the Social Security Number is unique on the EDB. Employees with invalid values of Social Security Number (all zeroes and all spaces) are also eliminated by the cursor. The cursor
is ordered by Social Security Number, which is the desired order for both the employee update and history files.

6. Finally, the campus location code (obtained from the Campus Control Record) is loaded into the location field of some work keys, and the interfaces to the various benefits modules to be called later are initialized.

**Employee Processing**

Employee processing is quite a simple process. Fetch the first employee from the EDB (using the SSN cursor), then begin a process of matching employees between the EDB and the employee history file, until all of the EDB employees have been processed. Finally, if any history file employees remain, they must be deleted from the IVR, since they are either no longer on the EDB or no longer eligible for inclusion on the IVR.

It should be noted here that although there are three separate files corresponding to the three PPS tables on the IVR database, it is only necessary to pass a delete record on the employee file to delete an employee from that database. It is not necessary to individually delete dependents and deductions from the IVR database.

The process of fetching an employee from the SSN cursor also involves fetching all of the required data from the PER table and filtering, based on those values, to see if there is interest in this employee for IVR purposes. Even if there is no interest, the employee must still be passed into the match logic because it might be necessary to delete the employee from the IVR due to an achieved separation or the employee going inactive.

The match logic simply compares the Social Security Number of the next EDB employee with the Social Security Number of the next history file employee. If the comparison indicates that the EDB employee precedes the history file employee, then the EDB employee is a candidate to be added to the IVR database. On the other hand, if the comparison indicates that the EDB employee follows the history file employee, the employee on the history file is either no longer on the EDB, or is still on the EDB but is no longer eligible for inclusion on the IVR, and should be deleted from the IVR. Finally, if the two sources reference the same employee, then it is necessary to check for any data changes. Once an employee is processed, the next employee from that particular source (either the EDB, the history file, or both) is moved into position to be compared. The real work of the program is contained in the processes which either add an employee to the IVR, delete an employee from the IVR, or check for and report any changes to an employee on the IVR.

**Employee Add Process:**

In the case of an add, there is no further selection to be done. It is necessary to obtain the rest of the employee-level data, complete the employee update record and employee history record; build and add both update and history records for each of the employee’s dependents; and build and add both update and history records for each of the employee’s significant deductions.
Completion of the employee-level data requires selection of necessary data from the PCM, PAY, BEN, BEL, and BND tables, to go along with data that has already been selected from the PER table. Many of the elements are taken just the way they appear on the EDB, others are altered in some way prior to being passed to the IVR, and still others are derived. Following is a summary of the cleanup or alterations which are done on the selected data elements:

All dates which are retrieved from the EDB with a DB/2 "low" value of 01/01/0001 are converted to the Sybase "low" value of 01/01/1900.

The Personnel Program corresponding to the employee's Primary Title Code is retrieved from the Title Code Table and passed to the IVR.

Foreign addresses are "packed" into the address space used for domestic addresses using an algorithm which attempts to maximize usage of the various foreign address elements which are available. For foreign addresses, the "state" and "zip code" fields are left unused. The additional fields "province", "country code" and "postal code" are retrieved from the FAD table. The "country code" is translated into a country name via lookup on the FCT table. The goal is to pack as much of this available information into the three fields: address line 1, address line 2, and city.

The first step in the packing algorithm is to see if the non-blank contents of both address line 1 and address line 2 can be gotten into address line 1. If that cannot be done, an attempt is made to get the city and country name into the city field. If that can be done, the address is left in that condition, so it looks like

Address line 1
Address line 2
City Country

If that cannot be done, then the contents of address line 2 are discarded, and the algorithm attempts to pack the four remaining fields (city, province, country name, and postal code) into address line 2 and city. Obviously, if any of these fields are blank, the packing is simplified. The target address format is

Address line 1
City Province
Country Postal Code

If all four fields cannot be accommodated, the postal code is given priority over the province, if the both are present. This gives addresses of the forms

Address line 1
City
Country Postal Code

Address line 1
City Province
Country

and
Depending upon which field(s) are present, if the postal code simply cannot be made to fit, the lowest priority acceptable address is:

Address line 1
City
Country

For each benefit plan in which the employee is enrolled, the employee cost and university contribution (as appropriate) are calculated by calling the appropriate PPBENx module. In addition, the following specific plan alterations are made:

For Group Life Insurance, the salary base is converted from being expressed in thousands of dollars to being expressed in dollars. Additionally, the total amount of insurance coverage under the Group Life plan is calculated and passed to the IVR.

For Accidental Death and Dismemberment, the principal sum is converted from being expressed in thousands of dollars to being expressed in dollars.

For UC Paid Life Insurance, the salary base is converted from being expressed in thousands of dollars to being expressed in dollars.

For Executive Life Insurance, the salary base is converted from being expressed in thousands of dollars to being expressed in dollars.

For each savings bond which the employee is purchasing, the denomination code is converted to the bond value before being passed to the IVR.

Processing of employee dependent records involves opening a cursor which selects all dependents for the employee, ordering them in dependent number order, and then generating an “add” update record and a history record for each dependent found. The data from the dependent rows is simply copied to the IVR, “low” dependent coverage effective dates are converted to “low” Sybase dates as was the case for dates within the employee record.

Processing of employee deduction records involves calling PPDSAUTL to load all of the DBL table rows for an employee, then walking through an internal table which identifies the deductions in which the IVR is interested and places them in the proper group for IVR processing. Based on the group, the program tests to see whether or not the employee has non-zero balance(s) which indicate that the deduction should be passed to the IVR. For each such deduction, an “add” update record and a history record is generated.
**Employee Delete Process:**

In the case of an employee delete, all that is necessary is to bypass each dependent and deduction history record for that employee, which effectively deletes them from the history file, generate a “delete” update record at the employee level, and bypass the employee history record, which effectively deletes it from the history file.

**Employee Change Process:**

The employee change process is complicated by the fact that we may or may not have to even look further for changes. Recall that during initialization, the current date obtained from the System Control Record was compared with a value contained in the employee history file header record. That comparison set a flag which indicated whether or not it would be necessary to check each employee for changes or whether we could use the update and compute timestamps to see if checking was necessary. The process begins by retrieving the PCM table data, which includes the last compute date. A flag indicating whether this employee has changed or not is then set. If all employees are being checked, the flag is simply set to indicate that the employee has changed. If individual employees are being checked, the flag is set to true or false based upon whether the update and compute timestamps have changed or not. Note that this flag does not necessarily indicate that changes which must be reported to the IVR have been made.

At this point, if the change flag is not set, there is no need to look at this employee any further. All that must be done is to copy the data from the input history record to the output history record and write the history record. Additionally, all of this employee’s dependent and deduction history records will be copied from the input history files to the output history files.

On the other hand, if the change flag is set, it is necessary to retrieve and format all of the employee-level data for this employee, just as was done in the case of an employee “add”. Once the data is available, the current values can be compared with the values recorded in the input history record, to see if reportable changes have occurred. If no such changes have been made, the input history data is simply copied to the output history data, otherwise, a “change” update record is generated and the updated employee data is moved to the output history record. Note that all employee-level data is reported on the change record; there is no need to do a field-by-field identification of changes.

Whether or not there were changes to the employee-level data, since the employee is being checked for changes, it is possible that there have been changes to either the dependent or deduction records which must be sent to the IVR as updates. Each of these two areas is processed in a similar fashion to the match/merge process that is used for employee-level data. Records corresponding to current EDB data are generated, in order, and compared with records in the dependent and deduction history files. As appropriate, “add,” “delete,” and “change” update records are generated, and the history records are added, updated, or deleted.
create table pps_emp (  
  emp_adc char(1),  
  emp_location char(2),  
  emp_ssn char(9),  
  emp_id_no char(9),  
  emp_name char(26),  
  emp_birthdate datetime,  
  emp_sex char(1),  
  emp_hire_date smalldatetime,  
  emp_home_dept char(6),  
  emp_loa_begin smalldatetime null,  
  emp_loa_return smalldatetime null,  
  emp_loa_type char(2),  
  emp_separate_date smalldatetime null,  
  emp_status_code char(1),  
  emp_primary_title char(4),  
  emp_personnel_prog char(1),  
  emp_fica_code char(1),  
  emp_FWT_gross decimal(9,2),  
  emp_address_1 char(30),  
  emp_address_2 char(30),  
  emp_address_city char(30),  
  emp_address_state char(2),  
  emp_address_zip char(5),  
  emp_retire_system char(1),  
  emp_ucpd_dis char(1),  
  emp_dcp_plan char(1),  
  emp_age_jan1 smallint,  
  emp_403b_change smalldatetime null,  
  emp_tip_code char(1),  
  emp_epd_wait_per smallint,  
  emp_epd_salary decimal(7,2),  
  emp_epd_pcmed smalldatetime null,  
  emp_epd_cost decimal(5,2),  
  emp_dent_plan char(2),  
  emp_dent_cov char(3),  
  emp_dent_pcmed smalldatetime null,  
  emp_dent_contri decimal(7,2),  
  emp_dent_cost decimal(7,2),  
  emp_gli_salary decimal(6,0),  
  emp_gli_plan char(1),  
  emp_gli_pcmed smalldatetime null,  
  emp_gli_ins_amt decimal(7,0),  
  emp_gli_cost decimal(7,2),  
  emp_dli_plan char(1),  
  emp_dli_pcmed smalldatetime null,  
  emp_dli_cost decimal(7,2),  
  emp_add_prin_sum decimal(6,0),  
  emp_add_cov char(1),  
  emp_add_pcmed smalldatetime null,  
  emp_add_cost decimal(7,2),  
  emp_med_plan char(2),  
  emp_med_cov char(3),  
  emp_med_pcmed smalldatetime null,  
  emp_med_contri decimal(7,2),  
  emp_med_cost decimal(7,2),  
  emp_ucpd_life_sal decimal(6,0),  
  emp_ucpd_life_pcmed smalldatetime null,  
  emp_opt_plan char(2),  
  emp_opt_cov char(3),  
  emp_opt_pcmed smalldatetime null,  
  emp_opt_contri decimal(7,2),  
  emp_opt_cost decimal(7,2)  
)
,emp_legal_plan    char(2)
,emp_legal_cov     char(3)
,emp_legal_pced    smalldatetime null
,emp_legal_contri   decimal(7,2)
,emp_legal_cost    decimal(7,2)
,emp_exec_life_ind char(1)
,emp_exec_life_sal  decimal(6,0)
,emp_exec_life_pced smalldatetime null
,emp_med_opt_out   char(1)
,emp_dent_opt_out  char(1)
,emp_opt_out       char(1)
,emp_pay_schedule  char(2)
,emp_beli_assigned char(1)
,emp_beli_derived  char(1)
,emp_beli_eff_date smalldatetime null
,emp_bond_1        decimal(4,0)
,emp_bond_2        decimal(4,0)
,emp_bond_3        decimal(4,0)
,emp_bond_4        decimal(4,0)

,constraint pps_emp_key
  primary key (emp_location,
              emp_ssn)
create table pps_dep (  
  dep_adc               char(1)  
  ,dep_location         char(2)  
  ,dep_emp_ssn          char(9)  
  ,dep_no               smallint  
  ,dep_name             char(26)  
  ,dep_birthdate        datetime  
  ,dep_relationship     char(1)  
  ,dep_ssn              char(9)  
  ,dep_sex              char(1)  
  ,dep_disabled         char(1)  
  ,dep_med_pced         smalldatetime null  
  ,dep_dent_pced        smalldatetime null  
  ,dep_opt_pced         smalldatetime null  
  ,dep_legal_pced       smalldatetime null  
  ,constraint pps_dep_key  
    primary key (dep_location,  
      dep_emp_ssn,  
      dep_no)  
  ,constraint pps_emp_dep  
    foreign key (dep_location,  
      dep_emp_ssn)  
    references pps_emp (emp_location,  
      emp_ssn)  
  )
create table pps_ded (  
  ded_adc         char(1)  
  ,ded_location    char(2)  
  ,ded_emp_ssn     char(9)  
  ,ded_group       char(1)  
  ,ded_gtn         char(3)  
  ,ded_g_balance   decimal(9,2)  
  ,ded_d_balance   decimal(9,2)  
  ,ded_y_balance   decimal(9,2)  
  ,ded_eff_date    smalldatetime null  
  ,constraint pps_ded_key  
    primary key (ded_location,  
                ded_emp_ssn,  
                ded_group,  
                ded_gtn)  
  ,constraint pps_emp_ded  
    foreign key (ded_location,  
                 ded_emp_ssn)  
    references pps_emp (emp_location,  
                         emp_ssn)  
)
<table>
<thead>
<tr>
<th>LINE</th>
<th>FIELD NAME</th>
<th>FIELD DESCRIPTION</th>
<th>DATA TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000100*</td>
<td><strong>01</strong></td>
<td><strong>IVREM-EMPLOYEE-RECORD.</strong></td>
<td><strong>CPWSIVRE</strong></td>
</tr>
<tr>
<td>001000</td>
<td>05</td>
<td><strong>IVREM-ADG-CODE</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>001100</td>
<td>88</td>
<td><strong>IVREM-BEING-ADDED</strong></td>
<td><strong>VALUE 'A'.</strong></td>
</tr>
<tr>
<td>001200</td>
<td>88</td>
<td><strong>IVREM-BEING-CHANGED</strong></td>
<td><strong>VALUE 'C'.</strong></td>
</tr>
<tr>
<td>001300</td>
<td>88</td>
<td><strong>IVREM-BEING-DELETED</strong></td>
<td><strong>VALUE 'D'.</strong></td>
</tr>
<tr>
<td>001400</td>
<td>05</td>
<td><strong>IVREM-KEY.</strong></td>
<td><strong>CPWSIVRE</strong></td>
</tr>
<tr>
<td>001500</td>
<td>10</td>
<td><strong>IVREM-LOCATION</strong></td>
<td><strong>PIC X(02).</strong></td>
</tr>
<tr>
<td>001600</td>
<td>10</td>
<td><strong>IVREM-SSN</strong></td>
<td><strong>PIC X(09).</strong></td>
</tr>
<tr>
<td>001700</td>
<td>05</td>
<td><strong>IVREM-DATA.</strong></td>
<td><strong>CPWSIVRE</strong></td>
</tr>
<tr>
<td>001800</td>
<td>10</td>
<td><strong>IVREM-ID-NO</strong></td>
<td><strong>PIC X(09).</strong></td>
</tr>
<tr>
<td>001900</td>
<td>10</td>
<td><strong>IVREM-NAME</strong></td>
<td><strong>PIC X(26).</strong></td>
</tr>
<tr>
<td>002000</td>
<td>10</td>
<td><strong>IVREM-BIRTH-DATE</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>002100</td>
<td>10</td>
<td><strong>IVREM-SEX</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>002200</td>
<td>10</td>
<td><strong>IVREM-HIRE-DATE</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>002300</td>
<td>10</td>
<td><strong>IVREM-HOME-DEPT</strong></td>
<td><strong>PIC X(06).</strong></td>
</tr>
<tr>
<td>002400</td>
<td>10</td>
<td><strong>IVREM-LOA-BEGIN-DATE</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>002500</td>
<td>10</td>
<td><strong>IVREM-LOA-RETURN-DATE</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>002600</td>
<td>10</td>
<td><strong>IVREM-LOA-TYPE</strong></td>
<td><strong>PIC X(02).</strong></td>
</tr>
<tr>
<td>002700</td>
<td>10</td>
<td><strong>IVREM-SEP-DATE</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>002800</td>
<td>10</td>
<td><strong>IVREM-EMP-STATUS</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>002900</td>
<td>10</td>
<td><strong>IVREM-DATE</strong></td>
<td><strong>PIC X(04).</strong></td>
</tr>
<tr>
<td>003000</td>
<td>10</td>
<td><strong>IVREM-PERSONNEL-PROG</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>003100</td>
<td>10</td>
<td><strong>IVREM-FICA-CODE</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>003200</td>
<td>10</td>
<td><strong>IVREM-FWT-GROSS</strong></td>
<td><strong>PIC X(07).</strong></td>
</tr>
<tr>
<td>003300</td>
<td>10</td>
<td><strong>IVREM-ADDRESS-1</strong></td>
<td><strong>PIC X(30).</strong></td>
</tr>
<tr>
<td>003400</td>
<td>10</td>
<td><strong>IVREM-ADDRESS-2</strong></td>
<td><strong>PIC X(30).</strong></td>
</tr>
<tr>
<td>003500</td>
<td>10</td>
<td><strong>IVREM-ADDRESS-CITY</strong></td>
<td><strong>PIC X(30).</strong></td>
</tr>
<tr>
<td>003600</td>
<td>10</td>
<td><strong>IVREM-ADDRESS-STATE</strong></td>
<td><strong>PIC X(02).</strong></td>
</tr>
<tr>
<td>003700</td>
<td>10</td>
<td><strong>IVREM-ADDRESS-ZIP</strong></td>
<td><strong>PIC X(05).</strong></td>
</tr>
<tr>
<td>003800</td>
<td>10</td>
<td><strong>IVREM-REIREMNT-CODE</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>003900</td>
<td>10</td>
<td><strong>IVREM-UCPD-DISABILITY</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>004000</td>
<td>10</td>
<td><strong>IVREM-AGE-JAN</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>004100</td>
<td>10</td>
<td><strong>IVREM-AGE-FEB</strong></td>
<td><strong>PIC X(02).</strong></td>
</tr>
<tr>
<td>004200</td>
<td>10</td>
<td><strong>IVREM-AGE-MAR</strong></td>
<td><strong>PIC X(03).</strong></td>
</tr>
<tr>
<td>004300</td>
<td>10</td>
<td><strong>IVREM-AGE-APR</strong></td>
<td><strong>PIC X(04).</strong></td>
</tr>
<tr>
<td>004400</td>
<td>10</td>
<td><strong>IVREM-AGE-MAY</strong></td>
<td><strong>PIC X(05).</strong></td>
</tr>
<tr>
<td>004500</td>
<td>10</td>
<td><strong>IVREM-EPD-WAIT-PERIOD</strong></td>
<td><strong>PIC X(03).</strong></td>
</tr>
<tr>
<td>004600</td>
<td>10</td>
<td><strong>IVREM-EPD-SALARY</strong></td>
<td><strong>PIC X(05).</strong></td>
</tr>
<tr>
<td>004700</td>
<td>10</td>
<td><strong>IVREM-EPD-PCEP</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>004800</td>
<td>10</td>
<td><strong>IVREM-EPD-COST</strong></td>
<td><strong>PIC X(03).</strong></td>
</tr>
<tr>
<td>004900</td>
<td>10</td>
<td><strong>IVREM-DENTAL-PLAN</strong></td>
<td><strong>PIC X(02).</strong></td>
</tr>
<tr>
<td>005000</td>
<td>10</td>
<td><strong>IVREM-DENTAL-COVERAGE</strong></td>
<td><strong>PIC X(03).</strong></td>
</tr>
<tr>
<td>005100</td>
<td>10</td>
<td><strong>IVREM-DENTAL-CONTR</strong></td>
<td><strong>PIC X(05).</strong></td>
</tr>
<tr>
<td>005200</td>
<td>10</td>
<td><strong>IVREM-DENTAL-COST</strong></td>
<td><strong>PIC X(05).</strong></td>
</tr>
<tr>
<td>005300</td>
<td>10</td>
<td><strong>IVREM-GLI-SALARY</strong></td>
<td><strong>PIC X(06).</strong></td>
</tr>
<tr>
<td>005400</td>
<td>10</td>
<td><strong>IVREM-GLI-PLAN-CODE</strong></td>
<td><strong>PIC X(01).</strong></td>
</tr>
<tr>
<td>005500</td>
<td>10</td>
<td><strong>IVREM-GLI-PCEP</strong></td>
<td><strong>PIC X(10).</strong></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>PIC</td>
<td>Code</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>005600</td>
<td>IVERMP-GL1-INS-AMOUNT</td>
<td>9(07)</td>
<td>005700</td>
</tr>
<tr>
<td>005800</td>
<td>IVERMP-GL1-INS-AMOUNT</td>
<td></td>
<td>005900</td>
</tr>
<tr>
<td>006000</td>
<td>IVERMP-DLI-PCED</td>
<td>X(10)</td>
<td>006100</td>
</tr>
<tr>
<td>006200</td>
<td>IVERMP-ADD-COVERAGE</td>
<td>X(01)</td>
<td>006300</td>
</tr>
<tr>
<td>006400</td>
<td>IVERMP-ADD-COST</td>
<td>9(05).99</td>
<td>006500</td>
</tr>
<tr>
<td>006600</td>
<td>IVERMP-MEDICAL-PLAN</td>
<td>X(03)</td>
<td>006700</td>
</tr>
<tr>
<td>006800</td>
<td>IVERMP-MEDICAL-PCED</td>
<td></td>
<td>006900</td>
</tr>
<tr>
<td>007000</td>
<td>IVERMP-MEDICAL-CONTRIB</td>
<td></td>
<td>007100</td>
</tr>
<tr>
<td>007200</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td>X(10)</td>
<td>007300</td>
</tr>
<tr>
<td>007400</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>007500</td>
</tr>
<tr>
<td>007600</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>007700</td>
</tr>
<tr>
<td>007800</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>007900</td>
</tr>
<tr>
<td>008000</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>008100</td>
</tr>
<tr>
<td>008200</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>008300</td>
</tr>
<tr>
<td>008400</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>008500</td>
</tr>
<tr>
<td>008600</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>008700</td>
</tr>
<tr>
<td>008800</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>008900</td>
</tr>
<tr>
<td>009000</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>009100</td>
</tr>
<tr>
<td>009200</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>009300</td>
</tr>
<tr>
<td>009400</td>
<td>IVERMP-UCPD-LIFE-PCED</td>
<td></td>
<td>009500</td>
</tr>
</tbody>
</table>
000100*******************************************************************************/ 28111012
000200* COPYMEMBER: CPWISIVRD */ 28111012
000300* RELEASE: 1012 SERVICE REQUEST(S): 12811 */ 28111012
000400* NAME: J.WILCOX CREATION DATE: 07/12/95 */ 28111012
000500* DESCRIPTION: RECORD LAYOUT FOR DEPENDENT UPDATE RECORD */ 28111012
000600* PASSED BETWEEN PPS AND THE BENCOM IVR SYSTEM. */ 28111012
000700*******************************************************************************/ 28111012
000800* CPWISIVRD
000900*01 IVRDEP-DEPENDENT-RECORD.
001000 05 IVRDEP-DEP-REC PIC X(01).
001100 88 IVRDEP-DEP-REC-VALUE PIC X(01).
001200 88 IVRDEP-DEP-REC-VALUE PIC X(01).
001300 88 IVRDEP-DEP-REC-VALUE PIC X(01).
001400 05 IVRDEP-LOC PIC X(02).
001500 10 IVRDEP-LOC-PIC PIC X(09).
001600 10 IVRDEP-LOC-SSN PIC X(09).
001700 10 IVRDEP-LOC-SSN PIC X(09).
001800 05 IVRDEP-LOC-SSN PIC X(09).
001900 10 IVRDEP-LOC-SSN PIC X(09).
002000 10 IVRDEP-LOC-SSN PIC X(09).
002100 10 IVRDEP-LOC-SSN PIC X(09).
002200 10 IVRDEP-LOC-SSN PIC X(09).
002300 10 IVRDEP-LOC-SSN PIC X(09).
002400 10 IVRDEP-LOC-SSN PIC X(09).
002500 10 IVRDEP-LOC-SSN PIC X(09).
002600 10 IVRDEP-LOC-SSN PIC X(09).
002700 10 IVRDEP-LOC-SSN PIC X(09).
002800 10 IVRDEP-LOC-SSN PIC X(09).

Attachment 5
000100************************************************************************** 28111012
000200* COPYMEMBER: CPWSIVRG * 28111012
000300* RELEASE: 1012 SERVICE REQUEST(S): 12811 * 28111012
000400* NAME: J.WILCOX CREATION DATE: 07/12/95 * 28111012
000500* DESCRIPTION: RECORD LAYOUT FOR DEDUCTION (GTN) RECORD * 28111012
000600* PASSED BETWEEN PPS AND THE BENCOM IVR SYSTEM. */ 28111012
000700************************************************************************** 28111012
000800* CPWSIVRG 28111012
000900*01 IVRDED-DEDUCTION-RECORD. CPWSIVRG
001000 05 IVRDED-ADC-CODE PIC X(01). CPWSIVRG
001100 88 IVRDED-BEING-ADDED VALUE 'A'. CPWSIVRG
001200 88 IVRDED-BEING-DELETED VALUE 'D'. CPWSIVRG
001300 88 IVRDED-BEING-CHANGED VALUE 'C'. CPWSIVRG
001400 05 IVRDED-KEY, CPWSIVRG
001500 10 IVRDED-LOCATION PIC X(02). CPWSIVRG
001600 10 IVRDED-EMP-SSN PIC X(09). CPWSIVRG
001700 10 IVRDED-GROUP-CODE PIC X(01). CPWSIVRG
001800 10 IVRDED-GTN-NUMBER PIC X(03). CPWSIVRG
001900 05 IVRDED-DATA, CPWSIVRG
002000 10 IVRDED-D-BALANCE PIC 9(07).99. CPWSIVRG
002100 10 IVRDED-D-BALANCE PIC 9(07).99. CPWSIVRG
002200 10 IVRDED-Y-BALANCE PIC 9(07).99. CPWSIVRG
002300 10 IVRDED-EFFECTIVE-DATE PIC X(10). CPWSIVRG
PPIIVR

"All" Run

Figure 1
PPIIVR

"Update" Run

![Diagram showing PPIIVR and its connections to other components like EDB, CTL, Employee History, Dependent History, Deduction History, Employee Update, Dependent Update, and Deduction Update.]

Figure 2
DATE MANDATED RELEASE

PAYROLL COORDINATION
(510) 987-0909
patty.yamashita@ucmva.ucop.edu

Date: August 22, 1995

To: Payroll Managers

From: Patty Yamashita

Subject: Release 1012, Benefits Interactive Voice Response (IVR) Project - bencom, Extract Process

Please find attached Release 1012 that addresses Service Request 12811. Service Request 12811 requested the development of an extract program that supplies data from the Payroll/Personnel System to the bencom (IVR) system. The extract program gathers data into three update files, which are the Employee Update File, the Dependent Update File, and the Deduction Update File.

This technical release is Date Mandated. Please coordinate the installation of this release with your technical group to ensure that UCOP receives the three update files by 5:00 P.M. on August 31, 1995.

If you have any comments or questions; please send e-mail or call.

cc: (without attachments)
    ssd, klk, mto, jcw, jsx