Individual Identification Numbers

FINAL

September 30th, 1993

Information Systems & Computing
Office of the President
University of California
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>DETAIL DESIGN - Individual ID Subsystem</td>
<td>2</td>
</tr>
<tr>
<td>Database Specification</td>
<td>2</td>
</tr>
<tr>
<td>Table IIDIDB.</td>
<td>3</td>
</tr>
<tr>
<td>Table IIDIXB.</td>
<td>3</td>
</tr>
<tr>
<td>Subroutine Modules</td>
<td>5</td>
</tr>
<tr>
<td>PROGRAM UCID100.</td>
<td>5</td>
</tr>
<tr>
<td>PROGRAM UCID200.</td>
<td>11</td>
</tr>
<tr>
<td>PROGRAM UCIDASN.</td>
<td>15</td>
</tr>
<tr>
<td>PROGRAM UCIDMNT.</td>
<td>19</td>
</tr>
<tr>
<td>PROGRAM UCIIDNUM.</td>
<td>23</td>
</tr>
<tr>
<td>PROGRAM UCIIDCHK.</td>
<td>23</td>
</tr>
<tr>
<td>Online Functions</td>
<td>24</td>
</tr>
<tr>
<td>Function IDMN.</td>
<td>24</td>
</tr>
<tr>
<td>Function IDBR.</td>
<td>25</td>
</tr>
<tr>
<td>Function IDDS.</td>
<td>27</td>
</tr>
<tr>
<td>Function IDAS.</td>
<td>29</td>
</tr>
<tr>
<td>Function IDUP.</td>
<td>31</td>
</tr>
<tr>
<td>Batch Processes</td>
<td>35</td>
</tr>
<tr>
<td>Interface Specifications</td>
<td>36</td>
</tr>
<tr>
<td>&quot;Random&quot; ID Number Generation</td>
<td>43</td>
</tr>
<tr>
<td>Guidelines for Campuses Not Using Random ID Numbers</td>
<td>43</td>
</tr>
<tr>
<td>Check Digit Algorithm</td>
<td>45</td>
</tr>
<tr>
<td>Partial Match Condition</td>
<td>46</td>
</tr>
</tbody>
</table>
Introduction

The development of Distributed On-line EDB Entry/Update has created the need for a mechanism by which Employee Identification Numbers are automatically assigned. Presently, Employee ID number assignment varies from campus to campus. Some campuses use the employee's Social Security Number as an Employee ID. Other campuses assign an "arbitrary" number, based on different algorithms, to the Employee ID. Regardless of the method of assignment, Employee ID's are presently established in central administrative offices such as the Payroll Office.

With on-line EDB Entry/Update, New Hire and Re-Hire activity occurring in departments will require assignment of an Employee ID. In order to insure that Employee ID assignment associated with such activity is consistent and controlled, an automated process for assigning Employee ID must be provided for use with on-line EDB Entry/Update. This document presents a design for such a mechanism.

The algorithm for the assignment of Employee ID presented in this document is that of a "random number generator". While this algorithm may not be desirable for all campuses, the design describes functions and features of an Individual Identification mechanism which will be valuable in the context of EDB Entry/Update. The design also presents a structure which supports campus flexibility in the implementation of the ID assignment mechanism. It is expected that campuses will modify the Employee ID assignment algorithm to meet local requirements (refer to "Guidelines for Campuses Not Using Random ID Numbers" later in this document).
DETAIL DESIGN - Individual ID Subsystem

This section describes the proposed ID Subsystem, the architecture, its database, and its relationship with external processes.

The primary goal of this subsystem is the management of identification numbers to new or rehired employees. This is accomplished by providing a repository of identity information, and a set of modules that operate against this repository to furnish the functions required of the system.

Although CICS inquiry screens will be supplied as a part of this system, any inquiry activity from other systems will be written as a part of that calling system. Inquiry from a variety of sources will be permitted in the ID Subsystem. Such inter-program inquiry may be made either by calling the IID100 module (described below) or by directly searching the ID database. No search restrictions are imposed.

Update of the ID Subsystem will, however, be strictly controlled and may take place only within this subsystem. Specifically, all updates must occur through IID100. This insures that consistent rules are applied to ID number assignment.

The assignment of ID will occur in a single callable module. This module will be a "dual-use" program, callable from an online CICS program or from a batch program. This module will manage the ID database, execute inquiries based on a variety of criteria, determine and assign new ID numbers, and calculate and verify check digits.
Database Specification

Table IIDIDB

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIDDB_INDIVIDUAL_ID</td>
<td>9(09) Indexed (unique)</td>
</tr>
<tr>
<td>IIDDB_FIRST_NAME</td>
<td>X(30) Varchar</td>
</tr>
<tr>
<td>IIDDB_MIDDLE_NAME</td>
<td>X(30) Varchar</td>
</tr>
<tr>
<td>IIDDB_LAST_NAME</td>
<td>X(30) VarChar, Indexed</td>
</tr>
<tr>
<td>IIDDB_SUFFIX</td>
<td>X(04)</td>
</tr>
<tr>
<td>IIDDB_BIRTHDATE</td>
<td>X(10) ISO, Indexed</td>
</tr>
<tr>
<td>IIDDB_SSN</td>
<td>X(09) Indexed (Unique if non-blank)</td>
</tr>
<tr>
<td>IIDDB_STATUS</td>
<td>X(01) (A=Active, P=Pending, N=Not In Use, R=Referred)</td>
</tr>
<tr>
<td>IIDDB_REF_NO</td>
<td>9(09) Foreign Key to IIDDB_INDIVIDUAL_ID</td>
</tr>
<tr>
<td>IIDDB_INACTIVE_DATE</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>IIDDB_INIT_ASSIGNED_TIMESTAMP</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>IIDDB_CHANGED_BY</td>
<td>X(08)</td>
</tr>
<tr>
<td>IIDDB_CHANGED_AT</td>
<td>TIMESTAMP</td>
</tr>
</tbody>
</table>

Table IIDIXB

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIDXB_INDIVIDUAL_ID</td>
<td>9(09) Concatenated key</td>
</tr>
<tr>
<td>IIDXB_SYSTEM_NAME</td>
<td>X(02) Concatenated key</td>
</tr>
<tr>
<td>IIDXB_SYSTEM_INDIVIDUAL_ID</td>
<td>X(09)</td>
</tr>
<tr>
<td>IIDXB_INIT_ASSIGNED_TIMESTAMP</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>IIDXB_CHANGED_BY</td>
<td>X(08)</td>
</tr>
<tr>
<td>IIDXB_CHANGED_AT</td>
<td>TIMESTAMP</td>
</tr>
</tbody>
</table>

1 The three columns of NAME will accept alphabetic characters in mixed UPPER/lower case. Any tabular display or report that lists detail in name sorted order will be case insensitive. Since the name is a combination of three columns, displays and reports will assemble the name to be shown. These columns are of varying length, up to 30 characters each. Standards may need to be established that specify how unusually long names are to be displayed.
Name suffix (e.g., Jr., III, M.D.) is kept separate from the name. This column is established when the ID number is issued or when the name modified. Searches against the database shall not consider an individual's name suffix. Accordingly, arguments passed for a database search should not contain any suffix information.

Suffix should be used primarily when browsing the database to assist in distinguishing one individual from another.

Since SSN is a basic identifier for an individual, indexing is required for this column. In some cases, however, SSN is not available at the time when the row is established. In that event, the row would be established with nulls in that column.
Subroutine Modules

PROGRAM UCIID100

UCIID100 is a module that performs a set of functions for a calling program against the IDB database. These functions, and the responses from UCIID100 are listed below:

1 Verify ID. Select a row in the IDB where ID matches the supplied ID number.

1.1 Supplied Parameters
1.1.1 ID number

1.2 Returned Values

1.2.1 Results of Verification
1.2.1.1 ID Found
1.2.1.2 ID Not Found
1.2.1.3 Invalid Check Digit
1.2.1.4 ID Valid, but referred to new number
1.2.2 Name (if found)
1.2.3 Name Suffix (if found)
1.2.4 Birthdate (if found)
1.2.5 SSN (if found)
1.2.6 Status flag (if found)
1.2.7 Referred ID (if ID is referred)
1.2.8 Inactive date (if ID is inactive or referred)
1.2.9 Date/time initially assigned
1.2.10 Last changed by
1.2.11 Last changed date/time

1.3 Processing.

1.3.1 Determine check digit validity
1.3.2 Select a row based on ID provided.
1.3.3 If IID-STATUS is equal to "R", set the "ID Referred to new number" flag
1.3.4 Set ID Found/Not Found indicator based on results of above select.

2 Find ID. Given birthdate, name, and/or SSN, search the IDB for a match or a partial match.
2.1 Supplied Parameters -- At least one of the following parameters must be supplied to UCIID100.

2.1.1 Birthdate (optional)
2.1.2 SSN (optional)
2.1.3 Name (optional)

2.2 Returned Values

2.2.1 Search result indicator
   2.2.1.1 Positive Match
   2.2.1.2 Partial Match
   2.2.1.3 No Match

2.2.2 List that contains one occurrence on positive match, zero occurrences on no match, and one or more occurrences on partial match.
   2.2.2.1 ID number
   2.2.2.2 Name
   2.2.2.3 SSN
   2.2.2.4 Birth date
   2.2.2.5 Status
   2.2.2.6 Referred ID
   2.2.2.7 Inactive Date
   2.2.2.8 Date/time initially assigned
   2.2.2.9 Last changed by
   2.2.2.10 Last changed date/time

2.3 Processing

2.3.1 If both birthdate and SSN have been supplied, search the IDB for full match on birthdate and SSN. Ignore records where the status indicator is set to "R" (referred). If found, return a positive match indicator. If no match is found, return a not found indicator.

2.3.2 If the birthdate or the SSN or the name has been supplied, then obtain the union of the following searches. If the result of this union is a null set, then return a not found indicator; otherwise return a partial match indicator as well as a list of all rows in this union. If the number of rows found exceed the size of the table being returned, limit the number of rows to the size of the table:
2.3.2.1 If the birthdate has been supplied, search the IDB for any rows matching that birthdate.

2.3.2.2 If the SSN has been supplied, search the IDB for any row matching that SSN.

2.3.2.3 If the last name or partial last name has been supplied, search the IDB for any row(s) where the first n characters of the last name match the n characters supplied in last name or partial last name⁴.

3 Update IDB row. Given ID number, Operator USERID⁵, update the indicated IDB row with the new data. **Note: It is the calling program's responsibility to verify that the row to be updated exists in the IDB prior to calling IDB100 with an UPDATE. This may be done by using the VERIFY function. The calling program is also required to supply the Last Changed Date timestamp to positively identify the row to be updated.**

3.1 Supplied Parameters

3.1.1 ID number (required)
3.1.2 Operator USERID (required)
3.1.3 Last Changed Date (required)
3.1.4 One or more of the following must be non-blank:
   3.1.4.1 Name
   3.1.4.2 Suffix
   3.1.4.3 Birthdate
   3.1.4.4 SSN
   3.1.4.5 Status
   3.1.4.6 Referred number

3.2 Returned Values

3.2.1 Result indicator
   3.2.1.1 Update completed
   3.2.1.2 Update failure - Row does not exist
   3.2.1.3 Update failure - Timestamp mismatch
   3.2.1.4 Update failure - New SSN exists on IDB.

3.3 Processing

3.3.1 Select the row on the IDB equal to the
ID number supplied.

3.3.1.1 If the row is not found, set the not found return indicator, and return.

3.3.2 Compare the Last Changed Date parameter to the Last Changed Date from the IDB row found.

3.3.2.1 If the timestamps do not match, set the timestamp mismatch indicator, and return.

3.3.3 If an SSN is supplied in the parameter list, compare the SSN on the selected record with the SSN supplied. If these are different, it implies that the SSN is being changed. If so, perform the following:

3.3.3.1 Select any row where SSN is equal to the SSN supplied by the calling program.

3.3.3.1.1 If one is found, set the duplicate SSN indicator and return.

3.3.4 Change the columns in the selected records with data from every non-blank field supplied by the calling program.

3.3.5 Set the IIDB_CHANGED_AT timestamp.

3.3.6 Update the row, set the update completed indicator, and return to the calling program.

4 Delete Row. Given ID number, delete the row matching this number. Note: It is the calling program's responsibility to verify that the row to be deleted exists in the IDB prior to calling IDB100 with an UPDATE. This may be done by using the VERIFY function. The calling program is also required to supply the Last Changed Date timestamp to positively identify the row to be deleted.

4.1 Supplied Parameters
4.1.1 ID Number (required)
4.1.2 Last Changed Date timestamp (required)

4.2 Returned Values

4.2.1 Deletion result indicator

4.2.1.1 Deletion complete
4.2.1.2 Deletion failure - timestamp mismatch
4.2.1.3 Deletion failure - row does not exist

4.3 Processing

4.3.1 Select the row on the IDB equal to the ID number supplied.

4.3.1.1 If the row is not found, set the not found return indicator, and return.

4.3.2 Compare the Last Changed Date parameter to the Last Changed Date from the IDB row found.

4.3.2.1 If the timestamps do not match, set the timestamp mismatch indicator, and return.

4.3.3 Delete the row with an ID number matching the ID number supplied.

5 ID Assignment. A minimum of birthdate and name are required to assign a new ID number in the IDB. If SSN or SSN and birthdate are supplied, then the IDB is searched to determine if either a full match exists on SSN and birthdate or a partial match exists on SSN alone. Under either of these cases, a return is made without assigning a new ID number. Otherwise, assign a new ID number, registering the identity information in the IDB, and return the new ID number.

5.1 Supplied Parameters

5.1.1 Birthdate (required)
5.1.2 SSN (optional)
5.1.3 Name (required)
5.1.4 Suffix (optional)
5.1.5 Operator USERID (required)

5.2 Returned Values

5.2.1 Assignment result indicator
   5.2.1.1 No ID assigned -- Positive match found
   5.2.1.2 No ID assigned -- duplicate SSN
   5.2.1.3 ID assigned

5.2.2 ID number assigned.
5.2.3 SSN (SSN supplied).
5.2.4 Birthdate (birthdate supplied).
5.2.5 Name (name supplied).
5.2.6 Suffix (suffix supplied)
5.2.7 Status (set to "A")
5.2.8 Referred number (as found on positive match)
5.2.9 Initial assignment timestamp
5.2.10 Changed-By (as supplied)
5.2.11 Changed-At (as established)

5.3 Processing

5.3.1 If an SSN was supplied, search the IDB for full match on birthdate and SSN. If found, return positive match.

5.3.2 If an SSN was supplied, determine if a row exists on the IDB which contains the same SSN. If found, return a duplicate SSN indicator.

5.3.3 Otherwise, generate an ID number, and create a row in the IDB database with supplied values.

Footnotes:

4 For example, an entry of "HO" in the last name supplied field would return a list of all last names in the IDB which either fully consist of "HO" or begin with "HO".

5 Operator USERID is needed to set the IIDB CHANGED BY column. If IID100 is being called from a batch program, USERID should be set to the name of the calling program.
PROGRAM UCIID200

UCIID200 is a module that performs a set of functions for a calling program against the IXB database. These functions, and the responses from UCIID200 are listed below:

1

Verify IXB Entry. Select a row in the IXB where the concatenated ID/system name matches the supplied ID number.

1.1 Supplied Parameters

1.1.1 ID number
1.1.2 System name

1.2 Returned Values

1.2.1 Results of Verification
1.2.1.1 Row Found
1.2.1.2 Row Not Found

1.2.2 ID Number found
1.2.3 System Name found
1.2.4 System Individual Id found
1.2.5 Initial Assignment timestamp found
1.2.6 Changed-by ID found
1.2.7 Changed-at timestamp found

1.3 Processing.

1.3.1 Select a row based on key provided.
1.3.2 Set row Found/Not Found indicator based on results of above select.

2

Add IXB row. Add a new row into the IXB. **Note:** It is the calling program's responsibility to verify that the row to be added does not already exist in the IXB prior to calling IDB200 with an ADD. This may be done by using the VERIFY function.

2.1 Supplied Parameters

2.1.1 IDB Identification number.
2.1.2 System identifier
2.1.3 System Identification number

2.2 Returned Values
2.2.1 Result of Add.
   2.2.1.1 Record Added
   2.2.1.2 Record not added; row with that IDB identification number and system identifier already exists.
   2.2.1.3 Record not added; IDB identification number does not exist in the IDB.

2.3 Processing.
   2.3.1 Verify that the ID number exists in the IDB. IF not, set the result indicator and return to the calling program.
   2.3.2 Select a row in the IXB that matches the key values provided. If one is found, set the result indicator and return to the calling program.
   2.3.3 Attempt to add record.
   2.3.4 Return indicating the status of the attempt.

Delete IXB row. Delete a row from the IXB. Note: It is the calling program's responsibility to verify that the row to be deleted exists in the IXB prior to calling IDB200 with an DELETE. This may be done by using the VERIFY function.

3.1 Supplied Parameters
   3.1.1 IDB Identification number.
   3.1.2 System identifier

3.2 Returned Values
   3.2.1 Result of Delete.
       3.2.1.1 Record Deleted
       3.2.1.2 Record not deleted; row with that key does not exist.

3.3 Processing.
   3.3.1 Select a row in the IXB that matches the values provided. If one is not found,
set the result indicator and return to the calling program

3.3.2 Delete the record.

3.3.3 Return indicating the status of the attempt.

4 Update IXB row. Change the system identification number for a row existing in the IXB. Note: It is the calling program's responsibility to verify that the row to be updated exists in the IDB prior to calling IDB200 with an UPDATE. This may be done by using the VERIFY function.

4.1 Supplied Parameters

4.1.1 IDB Identification number.
4.1.2 System identifier
4.1.3 System Identification number (value to be changed).

4.2 Returned Values

4.2.1 Result of Change.
   4.2.1.1 Record Changed
   4.2.1.2 Record not changed; row with that IDB identification number and system identifier does not exist.

4.3 Processing.
4.3.1 Select a row in the IXB that matches the values provided. If one is not found, set the result indicator and return to the calling program

4.3.2 Attempt to change record.

4.3.3 Return indicating the status of the attempt.

6This may be an IDB number equal to the key field of this
record, or it may be an ID number local to the system.
PROGRAM UCIIDASN

UCIIDASN is a module that performs the function of determining an identification number for an individual. This module does this by making inquiries to the IDB and/or IXB databases, creating a new number when appropriate, or indicating to the calling program that the operator must decide whether one or more partial matches belong to this individual. UCIIDASN calls other subroutines as needed to perform these functions. The requests made of this program and UCIIDASN's responses are listed below:

Note: Since this program initially issues a new ID number on a provisional basis, newly issued numbers in the IDB have the status indicator set to Pending. This pending indicator remains with the row until either a commit instruction is received from the calling program or a backout instruction is received. Calling this program with the backout function causes this subroutine to return the IDB and the IXB to their states prior to the number issuance.

This subroutine is required to save sufficient information when issuing an ID number to return these databases to their previous state.

1 Provide ID Number. (ASSN-UPDT) Provide an ID number to a calling program. This ID number may be either a newly generated ID number or an ID number that was previously issued to an individual having the same SSN and birthdate.

1.1 Supplied Parameters

1.1.1 Name (required, in 3-part format)
1.1.2 SSN (optional)
1.1.3 Birthdate (required)

1.2 Returned Values

1.2.1 Results of Action
   1.2.1.1 No match - Number issued
   1.2.1.2 Full match
   1.2.1.3 Partial match - resolution required

1.2.2 ID Number
1.3 Processing.

1.3.1 Match supplied Birthdate and SSN to the IDB

1.3.1.1 If fully matched:
   1.3.1.1.1 Update IDB name if it differs from the supplied name
   1.3.1.1.2 Set full match return value
   1.3.1.1.3 Return with the found ID

1.3.1.2 If partially matched:
   1.3.1.2.1 Set Partial match return value

1.3.1.3 If not matched:
   1.3.1.3.1 Generate a new ID number
   1.3.1.3.2 Insert IDB row with status flag set to pending
   1.3.1.3.3 Set partial match return value.

2 Resolved-Selected. (CONFIRMED-UPDT) This is the case where this program had previously returned to the calling program with a partial match return value after being called with a Provide ID function. A resolution has been made of the partial match, and a single record has been selected as the target of the assignment.

2.1 Supplied Parameters

   2.1.1 ID number selected
   2.1.2 Name (required, in 3-part format)
   2.1.3 SSN (optional)
   2.1.4 Birthdate (required)

2.2 Returned Values

   2.2.1 Results of Action
      2.2.1.1 Selected number assigned
      2.2.1.2 Unable to assign - SSN is duplicated

   2.2.2 ID Number

2.3 Processing.

   2.3.1 If supplied SSN not blank, select matching row from IDB that matches SSN

   2.3.1.1 If selected row found and ID not
equal to ID number selected:
  2.3.1.1.1 Set SSN-duplicated return value
  2.3.1.1.2 Return

2.3.2 Select row matching supplied ID number
  2.3.2.1 If row SSN differs from supplied SSN, update SSN
  2.3.2.2 If row birthdate differs from supplied birthdate, update birthdate
  2.3.2.3 If row name differs from supplied name, update name
  2.3.2.4 Set Selected-number assigned return value
  2.3.2.5 Return

Resolved-Issue. (CONFIRMED-ASSN) This is the case where this program had previously returned to the calling program with a partial match return value after being called with a Provide ID function. A resolution has been made that none of the partially matched records belongs to this individual.

3.1 Supplied Parameters
  3.1.1 Name (required, in 3-part format)
  3.1.2 SSN (optional)
  3.1.3 Birthdate (required)

3.2 Returned Values
  3.2.1 Results of Action
     3.2.1.1 Number assigned
     3.2.1.2 Unable to assign - SSN is duplicated
  3.2.2 ID Number

3.3 Processing.
  3.3.1 If supplied SSN not blank, select matching row from IDB that matches SSN
     3.3.1.1 If selected row found:
        3.3.1.1.1 Set SSN-duplicated return value
        3.3.1.1.2 Return
3.3.1.2 If selected row not found:
   3.3.1.2.1 Generate a new ID number
   3.3.1.2.2 Insert IDB row with status flag set to pending
   3.3.1.2.3 Set Number assigned return value.
   3.3.1.2.4 Return

4 Commit Pending Number. (PENDING-COMMIT) This call indicates that a number which was previously issued is now confirmed in the database of the calling system.

4.1 Supplied Parameters

4.1.1 ID Number previously issued
4.1.2 System Name
4.1.3 System Individual ID as known in the issuing system (in most cases equal to the ID number above, but it may differ)

4.2 Returned Values

4.2.1 Results of Action
   4.2.1.1 Number committed
   4.2.1.2 Unable to commit - ID does not exist

4.3 Processing.

4.3.1 Select row matching ID number previously issued ID number.

   4.3.1.1 If selected row found:
      4.3.1.1.1 Set status to 'A'
      4.3.1.1.2 Establish IXB row with supplied information. If one already exists, verify that the System Individual ID is equal to that supplied. If not, replace it with the one supplied.
      4.3.1.1.3 Return

   4.3.1.2 If selected row not found:
      4.3.1.2.1 Return with unable to commit--ID does not exist--return value set.

5 Backout Pending Number. (PENDING-DELETE) This call
indicates that the action which caused a number to be previously issued has been cancelled, and that any database entry previously made related to this issuance is to be retracted and the database returned to its former state.

5.1 Supplied Parameters

5.1.1 ID Number previously issued

5.2 Returned Values

5.2.1 Results of Action
5.2.1.1 Backout successful
5.2.1.2 Unable to backout - ID does not exist

5.3 Processing.

5.3.1 Retrieve saved database values. Delete any rows added, and restore any changes made.

PROGRAM UCIIDMNT

UCIIDMNT is a module that handles a change to any of the basic identifiers for an individual (i.e., ID Number, Name, SSN, and/or Birthdate).

Note: The previous program described, UCIIDASN makes changes on a provisional basis. Although this program is similar to UCIIDASN in that a change is made by calling UCIIDMNT with a "Change IDB data" function, and then with at "Commit IDB data" function, it differs from UCIIDASN in that minimal changes are made with the first call; substantive changes are made with the commit call. Nonetheless, this subroutine is required to save sufficient information when performing its functions to return these databases to their state previous to the change, should a backout be required.

1 Change IDB data. UCIIDMNT is called with record values to be changed in the IDB.

1.1 Supplied Parameters
1.1.1 ID Number prior to change (required)
1.1.2 ID Number after change (required only if ID number is changing; otherwise blank)
1.1.3 Name (required only if name changing, and then in 3-part format; otherwise blank)
1.1.4 SSN (required only if SSN is changing; otherwise blank)
1.1.5 Birthday (required only if birthdate is changing; otherwise blank)
1.1.6 Status (required only if Status is changing; otherwise blank)

1.2 Returned Values

1.2.1 Results of Action
1.2.1.1 Change authorized
1.2.1.2 Unable to change - ID number not found
1.2.1.3 Unable to change - Duplicate SSN

1.3 Processing.

1.3.1 Select row with new ID number (if new ID number not blank)
1.3.1.1 If not found, return with ID number not found return value.

1.3.2 Select row with new SSN (if new SSN not blank)
1.3.2.1 If found, return with Duplicate SSN return value.

1.3.3 Select row with previous ID number
1.3.3.1 If not found, return with ID number not found return value
1.3.3.2 If found return with change authorized return value.

2 Commit Pending Change. This call indicates that a change which was previously authorized is now confirmed in the database of the calling system.

2.1 Supplied Parameters

2.1.1 ID Number prior to change (required)
2.1.2 ID Number after change (required only if ID number is changing; otherwise blank)
2.1.3 Name (required only if name changing, and then in 3-part format; otherwise blank)
2.1.4 SSN (required only if SSN is changing; otherwise blank)
2.1.5 Birthdate (required only if birthdate is changing; otherwise blank)
2.1.6 Status (required only if Status is changing; otherwise blank)
2.1.7 System Name
2.1.8 System Individual ID as known in the issuing system (in most cases equal to the ID number above, but it may differ)

2.2 Returned Values

2.2.1 Results of Action
   2.2.1.1 Change committed
   2.2.1.2 Unable to commit - ID does not exist

2.3 Processing.

2.3.1 Select row matching ID number previously issued ID number.
   2.3.1.1 If selected row not found, return with unable to commit - id does not exist return value set.

   2.3.1.2 If new ID number not equal blank, select row matching new ID number. If selected row not found, return with unable to commit - id does not exist return value set.

   2.3.1.3 If this is an ID number change (new ID not blank) then:
      2.3.1.3.1 Set the referred ID on the old row to the new ID
      2.3.1.3.2 Set the status on the old row to R
      2.3.1.3.3 Set the SSN on the new row to the supplied SSN if the supplied SSN is not blank; otherwise set it to the SSN on the old row.
      2.3.1.3.4 Set status of the new row to 'A'
      2.3.1.3.5 Update both old and new rows
2.3.1.3.6 Remove any IXB row under the old ID number. Establish IXB row with new ID information.

2.3.1.3.7 Return

2.3.1.4 If this is other than an ID number change, then:
2.3.1.4.1 Make the changes as indicated in the input data.
2.3.1.4.2 Return

3 **Backout Pending Change.** This call indicates that a change which was previously authorized has been cancelled. Since the initial change processing was minimal, the backout processing is very straightforward.

3.1 Supplied Parameters

3.1.1 ID Number prior to change (required)
3.1.2 ID Number after change (required only if ID number is changing; otherwise blank)
3.1.3 Name (required only if name changing, and then in 3-part format; otherwise blank)
3.1.4 SSN (required only if SSN is changing; otherwise blank)
3.1.5 Birthdate (required only if birthdate is changing; otherwise blank)
3.1.6 Status (required only if Status is changing; otherwise blank)

3.2 Returned Values

3.2.1 Results of Action
3.2.1.1 Backout successful
3.2.1.2 Unable to backout - ID does not exist

3.3 Processing.

3.3.1 If new ID number not blank, then select row with new ID number
3.3.1.1 If not found, return with unable to backout - ID does not exist return value.
3.3.1.2 If found, then delete the row and return with the backout successful return value set.

3.3.2 If new ID number blank, then return with the backout successful return value set.

PROGRAM UCIIDNUM

UCIIDNUM is a module that generates a random number of varying length from six to nine digits as specified by the calling program.

1 Supplied Parameters
   1.1 None

2 Returned Values
   2.1 Number Generated

3 Processing.
   3.1 See "Random ID Number Generation" on page 44.

PROGRAM UCIIDCHK

UCIIDCHK is a module that, given a six digit ID number, calculates a check digit.

1 Supplied Parameters
   1.1 ID Number

2 Returned Values
   2.1 ID Number with check digit appended

3 Processing.
   3.1 See "Check Digit Algorithm" on page 45.
Online Functions

The Online portion of the Identification system is modeled after the online functions of the Payroll Personnel system. Screens are formatted in a similar fashion, and system navigation is comparable.

Direct access to the ID screens should be available from all Payroll system screens, and the CICS operator should be permitted to return to any Payroll system screen directly from an ID screen. This is accomplished by entering the function identifier.

Online functionality is provided through a series of callable subroutines which conform to, and take advantage of, services provided by UCRouter technology. UCRouter invokes, via subroutine call, the following service modules:

Function IDMN

Function IDMN will invoke program UCWFMNU to present a menu of options the Identification Number System. Choices available include:

- **IDBR** Browse Identification Database
- **IDDS** Display Individual Information
- **IDAS** Assign Identification Number
- **IDUP** Update Identification Information (includes deletion)

Selection criteria input fields are available on this screen so that the operator may uniquely identify an individual. This information is passed to the program controlling the selected option to indicate the requested individual.

1. **Screen fields**
   1.1 Next Function (input) Indicates next transaction to be executed.
   1.2 ID number (input)
Identification number to be operated on by the next function. This may be blank, a partial number, or a full number.

1.3 Name (input)
Name, if entered, should be in the form of LAST FIRST MIDDLE with optional commas between. This is the name to be operated on by the next function. This field may be blank, a partial name, or a full name.

1.4 SSN (input)
Social Security number to be operated on by the next function. This may be blank, a partial number, or a full number.

2 Active PF keys
2.1 PF1. Help
2.2 PF12. Quit

3 Transaction processing

3.1 Verify that the Next Function field is one of the permitted options. If it is, then transfer control to that transaction. Otherwise present an INVALID FUNCTION message, and re-invoke IMNMU

4 Control transfers

4.1 Control may be transferred to any transaction requested, including those which are not displayed in the options list, provided such transfer is permitted by MVS Resource Security.

Function IDBR

Function IDBR permits the operator to browse a tabular list of entries in the IDB. This list is a subset based on a selection criteria entered by the operator.

The screen layout is modeled on the Payroll system's IBRS screen where selection criteria fields are integrated with the tabular display screen. The selection criteria fields are: ID number, name, birthdate, and SSN. The system presents the screen with the cursor active on the first selection criteria field. Entry in one or more of the
selection criteria fields is mandatory. The system searches the database for positive or partial matches and displays the results.

The additional functions of obtaining additional information about an entry, correcting an entry, referring an identification number, or deleting an identification number (provided the operator is authorized for those functions) is available from the IDBR screen by entering the appropriate transaction name in the Next Function field and positioning the cursor next to the record to be processed.

1 Screen fields

1.1 List of Records (display)
   1.1.1 ID Number
   1.1.2 Name
   1.1.3 Birth date
   1.1.4 SSN

1.2 Next Function (input)
Indicates next transaction to be executed.

1.3 ID number (input)
Identification number to be operated on by the next function. This may be blank, a partial number, or a full number.

1.4 Name (input)
Name, if entered, should be in the form of LAST FIRST MIDDLE with optional commas between last and first name. This is the name to be operated on by the next function. This field may be blank, a partial name, or a full name.

1.5 SSN (input)
Social Security number to be operated on by the next function. This may be blank, a partial number, or a full number.

2 Active PF keys

2.1 PF1 Help
2.2 PF3 Previous Menu
2.3 PF7 Back (if there are prior entries in the list)
2.4 PF8 Forward (if there are following entries in the list)
2.5 PF9 Main Menu
2.6 PF12 Quit
3 Transaction processing

3.1 Determine if the cursor is on a list element. If so, save the ID number of that list element in the COMMAREA to be passed to the next transaction.

3.2 Verify that the Next Function field is one of the permitted options. If it is not, present an INVALID FUNCTION message, and re-invoke IDBR.

4 Control transfers

4.1 Control may be transferred to any transaction requested, including those which are not displayed in the options list, provided such transfer is permitted by MVS Resource Security.

Function IDDS

Function IDDS displays the row information for a particular entry in the IDB. It may be called from either IDMN or from IDBR. If called from IDBR, an ID number will be supplied to permit a direct database read and posting of information to the screen. If called from IDMN, an ID number may or may not be supplied. When an ID number is supplied, then the program executes a database read and displays the row information for the entry in the IDB table. If no ID number is supplied, the program positions the cursor on the selection criteria fields, and the operator is required to enter one or more fields. If the entered data is sufficient to identify an individual IDB entry, then the IDDS screen is redisplayed with the row entry, otherwise IDBR is invoked to list the records satisfying the selection criteria.

1 Screen fields

1.1 ID Number
1.2 Name
1.3 Birth date
1.4 SSN
1.5 Status
1.6 Referred ID
1.7 Next Function (input)
Indicates next transaction to be executed.
1.8 ID number (input)
Identification number to be operated on by the next function. This may be blank, a partial number, or a full number.

1.9 Name (input)
Name, if entered, should be in the form of LAST FIRST MIDDLE with optional commas between. This is the name to be operated on by the next function. This field may be blank, a partial name, or a full name.

1.10 SSN (input)
Social Security number to be operated on by the next function. This may be blank, a partial number, or a full number.

2 Active PF keys
2.1 PF1 Help
2.2 PF3 Previous Menu
2.3 PF9 Main Menu
2.4 PF12 Quit

3 Transaction processing

3.1 Determine if any data has been provided in the record selection fields. If ID number, Name, or SSN has been entered, then attempt to retrieve a row from the IDB based on the value of the supplied field. If a record is retrieved which fully matches the entered field, then re-invoke IDDS, displaying the fields of that record.

If no record can be found which fully matches the entered field, then set up the COMMAREA with the selection data, preset "IDDS" in the Next Function field, and transfer control to IDBR for record selection.

3.2 Verify that the Next Function field is one of the permitted options. If it is not, present an INVALID FUNCTION message, and re-invoke IDBR.

4 Control transfers

4.1 Control may be transferred to any transaction requested, including those which are not displayed in the options list, provided such transfer is
permitted by MVS Resource Security.

Function IDAS

Function IDAS supports the allocation of ID numbers. IDBS may be invoked from IDMN or from IDBR or from any other online screen.

The program permits input of the first/middle/last name fields, the birth date, and the SSN field. The minimum data required to establish a row in the IDB are the name and birth date fields. The operator is asked to complete identification information for the individual.

When the data has been entered, a search of the IDB is made for any record which fully or partially matches the entered information. If no full or partial matches are found, a new ID number is selected by the system, and the elemental information supplied is posted with that new ID number to the IDB.

Any records which are partial or full matches are displayed via the IDBR screen. The operator may then determine whether one of the partially matched records belongs to this individual.

If the operator determines that none of the partially matched records belongs to this individual, then the ID number assignment continues even though there are partial matches. This will proceed, upon operator confirmation, provided that such ID number assignment does not result in a duplicate SSN on the IDB.

1 Screen fields

1.1 ID Number (display)
1.2 Name (input) -- First Middle Last name fields.
1.3 Birth date (input)
1.4 SSN (input)
1.5 Next Function (input)
   Indicates next transaction to be executed.
1.6 ID number (input)
   Identification number to be operated on by the next function. This may be blank, a partial number, or a full number.
1.7 Name (input)
Name, if entered, should be in the form of LAST FIRST MIDDLE with optional commas between. This is the name to be operated on by the next function. This field may be blank, a partial name, or a full name.

1.8 SSN (input)
Social Security number to be operated on by the next function. This may be blank, a partial number, or a full number.

2 Active PF keys
2.1 PF1 Help
2.2 PF3 Previous Menu
2.3 PF9 Main Menu
2.4 PF12 Quit

3 Transaction processing

3.1 Present the IDAS screen until all elemental data has been entered. Construct a name field with the format of Last, First Middle. Use module IID100 to search for records which either fully or partially match the elemental data supplied. If any records are found that fully or partially match this record, pass control to IDBR to display those records.

If no record can be found which fully or partially matches the entered fields, then proceed with the ID number assignment by repeating the following steps until an acceptable number is found:

3.1.1 Randomly generate a six digit number.
3.1.2 Calculate a check digit based on the six digit number. Append the check digit, forming the seven digit ID number.
3.1.3 Read the IDB to determine if this ID number is being used. If not, this is an acceptable number.

Create a row in the IDB containing the elemental data supplied along with the newly generated ID number. Display the ID number. The creation of the IDB row should not result in a duplicate SSN.
3.2 Verify that the Next Function field is one of the permitted options. If it is not, present an INVALID FUNCTION message, and re-invoke IDUP.

4 Control transfers

4.1 Control may be transferred to any transaction requested, including those which are not displayed in the options list, provided such transfer is permitted by MVS Resource Security.

Function IDUP

Function IDUP performs multiple functions against the IDB. First, it may be used to change name, SSN, or birthdate. This may be done provided the new SSN does not duplicate any other SSN on the database. (Duplicate names or birth dates are permitted). Changes are accomplished by displaying the record and accepting changes to name, SSN, or birthdate. When the operator is ready to update the database, but before the actual update, a search is made of the IDB for any other record with the same SSN. If a SSN match is found, the update is aborted, and the operator is notified of a duplicate SSN situation. If no record is found matching the SSN, then the system notifies the operator of its intent to update and requests a confirmation. Upon confirmation, the update proceeds.

IDUP may also be used to refer an identification number. This would be required when an individual was assigned two or more identification numbers in error.

A referral is processed similar to an update. IDUP displays the row information for a particular entry in the IDB, and permits a change to the IIDB_STATUS and the IIDB_REF_NO columns. When an ID number is supplied, then the program executes a database read and displays the row information for the entry in the IDB table.

When a record is selected for referral, IIDB-INDIVIDUAL-STATUS column for the referred record (i.e., the obsolete record) is set to "R", and the identification number of the record being referred to is inserted in the IIDB-INDIVIDUAL-REF-NO column. IIDB-INDIVIDUAL-SSN is set to nulls. This combination of events signals both that the identification number is in use and that referral is made to the correct employee identification. It is required that IIDB_REF_NO
exist as a row in the IDB. When the system is prepared to
perform the update, the system notifies the operator of its
intent to update and requests a confirmation. Upon
confirmation, the update proceeds.

Finally, a record may be deleted from the IDUP screen. This
would be done when (1) an individual was assigned two or
more identification numbers in error, but (2) activity has
occurred only under one of the numbers. IDUP displays the
row information for a particular entry in the IDB, and
permits the operator to indicate that this row is to be
deleted by activating a unique PF function key.

A row may be deleted if the IIDB_STATUS column is not set to
"A" and no record exists where IIDB_REF_NO equals the
IIDB_INDIVIDUAL_ID of this row. If the row is determined to
be eligible for delete, and when the operator is ready to
delete the row, but before the actual update, the program
requests a delete confirmation from the operator. When the
operator confirms the delete, then an SQL DELETE RESTRICT
statement is issued. This will cause a delete to occur only
when no IIDB_REF_NO foreign key refers to the
IIDB_INDIVIDUAL_ID in this row. If this SQL statement
violates a referential constraint, then no delete action is
performed and the operator is notified of the situation.

1 Screen fields

1.1 ID Number (display)
1.2 Name (i/o)
1.3 Birth date (i/o)
1.4 SSN (i/o)
1.5 Status (i/o)
1.6 Referred ID (i/o)
1.7 Delete indicator (Y/N) (input)
1.8 Next Function (input)
   Indicates next transaction to be executed.

1.9 ID number (input)
   Identification number to be operated on by the
   next function. This may be blank, a partial
   number, or a full number.

1.10 Name (input)
   Name, if entered, should be in the form of LAST
   FIRST MIDDLE with optional commas between. This
   is the name to be operated on by the next
function. This field may be blank, a partial name, or a full name.

1.11 SSN (input)
Social Security number to be operated on by the next function. This may be blank, a partial number, or a full number.

2 Active PF keys
2.1 PF1 Help
2.2 PF3 Previous Menu
2.3 PF9 Main Menu
2.4 PF12 Quit

3 Transaction processing

3.1 Determine if any data has been provided in the record selection fields. If ID number, Name, or SSN has been entered, then attempt to retrieve a row from the IDB based on the value of the supplied field. If a record is retrieved which fully matches the entered field, then re-invoke IDUP, displaying the fields of that record.

If no record can be found which fully matches the entered field, then set up the COMMAREA with the selection data, preset "IDUP" in the Next Function field, and transfer control to IDBR for record selection.

3.2 Once control is returned from the operator following the display and possible update of IDB row data, the next steps should be taken to achieve the database update.

3.2.1 If the name has changed, confirm that the new name is not blank. If it has, re-invoke IDUP, displaying a message that indicates the name format and the minimum requirements.

3.2.2 If the SSN has changed, determine if this SSN is in the IDB. If it is, re-invoke IDUP, displaying a message that indicates that this new SSN creates a duplicate SSN.

3.2.3 Ascertaint that all other changes do not violate any edit rules and that the record is internally consistent.
3.2.4 When no other errors exist, re-invoke IDUP, displaying a message that indicates an intent to update, and requesting operator confirmation. Upon operator confirmation, proceed with the database update. If the operator fails to confirm, then abandon the update attempt.

3.2.5 If the operator has requested a record delete, the system determines whether the record exists. The operator is asked to confirm the deletion of the record. When the operator provides confirmation, the row is deleted. DB2 may not permit the deletion if any other row in the database is referring to that row as a foreign key in the Reference number field. If that does occur, then delete processing is stopped, and the operator is notified of the situation.

3.3 Verify that the Next Function field is one of the permitted options. If it is not, present an INVALID FUNCTION message, and re-invoke IDUP.

4 Control transfers

4.1 Control may be transferred to any transaction requested, including those which are not displayed in the options list, provided such transfer is permitted by MVS Resource Security.
Batch Processes

The single batch process required for the Individual Identification System would be an auditing procedure specific to each installation.

1. Process each row in the IXB. Verify that the IXB row has a corresponding record in the application's files. If it does not, then report the discrepancy.

2. Process each row in the IDB, ignoring rows where IIDB_STATUS is set to "R". Verify that at least one IXE row exists with a matching INDIVIDUAL_ID. If so, set the IIDB_STATUS to "C" (if it is not already set to that value). Otherwise set IIDB_STATUS to "N".
Interface Specifications

Any program calling UCIID100 must use the following structure to pass information to and from routine UCIID100. This is an EXTERNAL structure. (Copylib member UCIDWS10).

01 UCIDWS10-INTERFACE EXTERNAL.
  03 UCIDWS10-EXT-AREA PIC X(3000).
  03 UCIDWS10-EXT-DATA REDEFINES UCIDWS10-EXT-AREA.
    05 UCIDWS10-RETURN-STATUS PIC X.
       88 UCIDWS10-FATAL-ERROR VALUE 'Y'.
       88 UCIDWS10-NO-ERRORS VALUE 'N'.
    05 UCIDWS10-ERROR-PARA PIC X(04).
    05 UCIDWS10-ERROR-CODE PIC 9(04).
    05 UCIDWS10-CALL-REQ PIC X(06).
       88 UCIDWS10-REQ-VERIFY VALUE 'VRFY'.
       88 UCIDWS10-REQ-FIND VALUE 'FIND'.
       88 UCIDWS10-REQ-ASSIGN VALUE 'ASSN'.
       88 UCIDWS10-REQ-UPDATE VALUE 'UPD'.
       88 UCIDWS10-REQ-DELETE VALUE 'DEL'.
       88 UCIDWS10-REQ-BROWSE VALUE 'BROW'.
    05 UCIDWS10-RETURN-STAT PIC X(06).
       88 UCIDWS10-STAT-VRFY-FND VALUE 'VFOUND'.
       88 UCIDWS10-STAT-VRFY-NOT-FND VALUE 'VNTFND'.
       88 UCIDWS10-STAT-VRFY-BAD-CHKDT VALUE 'VBDCHK'.
       88 UCIDWS10-STAT-VRFY-REF VALUE 'VREFER'.
       88 UCIDWS10-STAT-FIND-FND VALUE 'FFOUND'.
       88 UCIDWS10-STAT-FIND-NOT-FND VALUE 'FNTFND'.
       88 UCIDWS10-STAT-FIND-LIST VALUE 'FDLIST'.
       88 UCIDWS10-STAT-ASSN-ASSND VALUE 'AASSND'.
       88 UCIDWS10-STAT-ASSN-NOT-ASSND VALUE 'ANTASN'.
       88 UCIDWS10-STAT-ASSN-DUP-SSN VALUE 'ADUPSNN'.
       88 UCIDWS10-STAT-UPD-UPDATED VALUE 'UDPDTD'.
       88 UCIDWS10-STAT-UPD-NOT-FND VALUE 'UNTFFND'.
       88 UCIDWS10-STAT-UPD-DUP-SSN VALUE 'UDUPSNN'.
       88 UCIDWS10-STAT-DEL-DELETED VALUE 'DDELTDT'.
       88 UCIDWS10-STAT-DEL-NOT-FND VALUE 'DNFTFND'.
    05 UCIDWS10-PASSED-FIELDS.
       10 UCIDWS10-PASSED-IID PIC X(09).
       10 UCIDWS10-PASSED-SSID PIC X(09).
       10 UCIDWS10-PASSED-BIRTHDATE PIC X(10).
       10 UCIDWS10-PASSED-NAME-LAST PIC X(30).
       10 UCIDWS10-PASSED-NAME-FIRST PIC X(30).
       10 UCIDWS10-PASSED-NAME-MIDDLE PIC X(30).
       10 UCIDWS10-PASSED-NAME-SUFFIX PIC X(04).
       10 UCIDWS10-PASSED-CHANGED-BY PIC X(08).
       10 UCIDWS10-PASSED-STAT PIC X(01).
       10 UCIDWS10-PASSED-REF-IID PIC X(09).
       10 UCIDWS10-PASSED-CHANGED-AT PIC X(26).
05 UCIDWS10-RECORD-RET-COUNT PIC S9(03) COMP-3.
05 UCIDWS10-TABLE-RETURNED.
10 UCIDWS10-TABLE-MAX-SIZE PIC S9(4) COMP.
10 UCIDWS10-TABLE-CURR-SIZE PIC S9(4) COMP.
10 UCIDWS10-RET-TABLE OCCURS 13 TIMES INDEXED BY UCIDWS10-RECORD-INDEX.
   15 UCIDWS10-RET-IID PIC X(09).
   15 UCIDWS10-RET-SSN PIC X(09).
   15 UCIDWS10-RET-BIRTHDATE PIC X(10).
   15 UCIDWS10-RET-NAME-LAST PIC X(30).
   15 UCIDWS10-RET-NAME-FIRST PIC X(30).
   15 UCIDWS10-RET-NAME-MIDDLE PIC X(30).
   15 UCIDWS10-RET-NAME-SUFFIX PIC X(04).
   15 UCIDWS10-RET-CHANGED-BY PIC X(08).
   15 UCIDWS10-RET-STATUS PIC X(01).
   15 UCIDWS10-RET-REF-IID PIC X(09).
   15 UCIDWS10-RET-ASSIGNED-AT PIC X(26).
   15 UCIDWS10-RET-CHANGED-AT PIC X(26).
   15 FILLER PIC X(08).
10 FILLER PIC X(207).
Any program calling UCIID200 must use the following structure to pass information to and from routine UCIID200. This is an EXTERNAL structure. (Copylib member UCIDWS20).

01 UCIDWS20-INTERFACE EXTERNAL.
   03 UCIDWS20-EXT-AREA PIC X(1000).
   03 UCIDWS20-EXT-DATA REDEFINES UCIDWS20-EXT-AREA.
      05 UCIDWS20-RETURN-STATUS PIC X(01).
      88 UCIDWS20-FATAL-ERROR VALUE 'Y'.
      88 UCIDWS20-NO-ERRORS VALUE 'N'.
      05 UCIDWS20-ERROR-PARA PIC X(04).
      05 UCIDWS20-ERROR-SQRCODE PIC 9(04).
      05 UCIDWS20-CALL-REQUEST PIC X(06).
      88 UCIDWS20-REQ-VERIFY VALUE 'VERIFY'.
      88 UCIDWS20-REQ-ADD VALUE 'ADD'.
      88 UCIDWS20-REQ-UPDATE VALUE 'UPDATE'.
      88 UCIDWS20-REQ-DELETE VALUE 'DELETE'.
      05 UCIDWS20-RETURN-STAT PIC X(06).
      88 UCIDWS20-STAT-VER-FOUND VALUE 'FOUND'.
      88 UCIDWS20-STAT-VER-NOT-FOUND VALUE 'NOFIND'.
      88 UCIDWS20-STAT-ADD-OK VALUE 'ADDOK'.
      88 UCIDWS20-STAT-ADD-DUP-ROW VALUE 'ADUPRO'.
      88 UCIDWS20-STAT-ADD-NOT-IN-IDB VALUE 'ANOIDB'.
      88 UCIDWS20-STAT-DEL-OK VALUE 'DELOK'.
      88 UCIDWS20-STAT-DEL-NOT-THERE VALUE 'DNOROW'.
      88 UCIDWS20-STAT-UPD-OK VALUE 'UPDTOK'.
      88 UCIDWS20-STAT-UPD-NOT-THERE VALUE 'UNOROW'.
      88 UCIDWS20-STAT-FATAL-ERROR VALUE 'ERROR'.
   05 UCIDWS20-PASSED-FIELDS.
      10 UCIDWS20-PASS-IID PIC X(09).
      10 UCIDWS20-PASS-SYSTEM-NAME PIC X(02).
      10 UCIDWS20-PASS-SYSTEM-IID PIC X(09).
      10 UCIDWS20-PASS-CHANGED-BY PIC X(08).
      10 UCIDWS20-PASS-CHANGED-AT PIC X(26).
   05 UCIDWS20-RETURNED-FIELDS.
      10 UCIDWS20-RET-IID PIC X(09).
      10 UCIDWS20-RET-SYSTEM-NAME PIC X(02).
      10 UCIDWS20-RET-SYSTEM-IID PIC X(09).
      10 UCIDWS20-RET-ASSIGNED-AT PIC X(26).
      10 UCIDWS20-RET-CHANGED-BY PIC X(08).
      10 UCIDWS20-RET-CHANGED-AT PIC X(26).
   05 FILLER PIC X(845).
Any program calling UCIIDASN must use the following structure to pass information to and from routine UCIIDASN. This is an EXTERNAL structure. (Copylib member UCIDWSAS).

01 UCIDWSAS-INTERFACE EXTERNAL.
  03 UCIDWSAS-EXT-AREA PIC X(500).
  03 UCIDWSAS-EXT-DATA REDEFINES UCIDWSAS-EXT-AREA.
    05 UCIDWSAS-RETURN-STATUS PIC X.
      88 UCIDWSAS-FATAL-ERROR VALUE 'Y'.
      88 UCIDWSAS-NO-ERRORS VALUE 'N'.
    05 UCIDWSAS-ERROR-PROG PIC X(08).
    05 UCIDWSAS-ERROR-PARA PIC X(04).
    05 UCIDWSAS-ERROR-CODE PIC 9(04).
    05 UCIDWSAS-CALL-REQ PIC X(06).
      88 UCIDWSAS-REQ-ASSN-UPDT VALUE 'ASNUPT'.
      88 UCIDWSAS-REQ-CONFIRMED-ASSN VALUE 'CNFSN'.
      88 UCIDWSAS-REQ-CONFIRMED-UPDT VALUE 'CNFUPT'.
      88 UCIDWSAS-REQ-PENDING-COMMIT VALUE 'PNDCMT'.
      88 UCIDWSAS-REQ-PENDING-DELETE VALUE 'PNDDEL'.
    05 UCIDWSAS-RETURN-STAT PIC X(06).
      88 UCIDWSAS-STAT-ASSIGNED VALUE 'ASSN '.
      88 UCIDWSAS-STAT-UPDATED VALUE 'UPDAT '.
      88 UCIDWSAS-STAT-RESOL-REQD VALUE 'RESREQ'.
      88 UCIDWSAS-STAT-DUP-SSN VALUE 'DUPSSN'.
      88 UCIDWSAS-STAT-PEND-COMMITTED VALUE 'PNDCMT'.
      88 UCIDWSAS-STAT-PEND-DELETED VALUE 'PNDDEL'.
      88 UCIDWSAS-STAT-NOT-FOUND VALUE 'NOTFND'.
      88 UCIDWSAS-STAT-BDERR VALUE 'BDERR '.
      88 UCIDWSAS-STAT-NAMERR VALUE 'NAMERR'.
    05 UCIDWSAS-PASSED-FIELDS.
      10 UCIDWSAS-PASSED-IID PIC X(09).
      10 UCIDWSAS-PASSED-SSN PIC X(09).
      10 UCIDWSAS-PASSED-BIRTHDATE PIC X(10).
      10 UCIDWSAS-PASSED-NAME-LAST PIC X(30).
      10 UCIDWSAS-PASSED-NAME-FIRST PIC X(30).
      10 UCIDWSAS-PASSED-NAME-MIDDLE PIC X(30).
      10 UCIDWSAS-PASSED-NAME-SUFFIX PIC X(04).
      10 UCIDWSAS-PASSED-CHANGED-BY PIC X(08).
      10 UCIDWSAS-PASSED-STAT PIC X(01).
      10 UCIDWSAS-PASSED-REF-IID PIC X(09).
      10 UCIDWSAS-PASSED-CHANGED-AT PIC X(26).
    05 UCIDWSAS-PASSED-SYS-FIELDS.
      10 UCIDWSAS-PASSED-SYS-NAME PIC X(02).
      10 UCIDWSAS-PASSED-SYS-IID PIC X(09).
    05 UCIDWSAS-RECORD-RET-COUNT PIC S9(03) COMP-3.
    05 UCIDWSAS-RETURNED-FIELDS.
      10 UCIDWSAS-RET-IID PIC X(09).
      10 UCIDWSAS-RET-SSN PIC X(09).
      10 UCIDWSAS-RET-BIRTHDATE PIC X(10).
      10 UCIDWSAS-RET-NAME-LAST PIC X(30).

39
10 UCIDWSAS-RET-NAME-FIRST          PIC X(30).
10 UCIDWSAS-RET-NAME-MIDDLE         PIC X(30).
10 UCIDWSAS-RET-NAME-SUFFIX         PIC X(04).
10 UCIDWSAS-RET-CHANGED-BY          PIC X(08).
10 UCIDWSAS-RET-STATUS              PIC X(01).
10 UCIDWSAS-RET-REF-IID             PIC X(09).
10 UCIDWSAS-RET-ASSIGNED-AT         PIC X(26).
10 UCIDWSAS-RET-CHANGED-AT          PIC X(26).
05 FILLER                           PIC X(100).
Any program calling UCDIMNT must use the following structure to pass information to and from routine UCDIMNT. This is an EXTERNAL structure. (Copylib member UCIDWSMT).

01 UCIDWSMT-INTERFACE EXTERNAL.
  03 UCIDWSMT-EXT-AREA PIC X(2000).
  03 UCIDWSMT-EXT-DATA REDEFINES UCIDWSMT-EXT-AREA.
    05 UCIDWSMT-RETURN-STATUS PIC X.
      88 UCIDWSMT-FATAL-ERROR VALUE 'Y'.
      88 UCIDWSMT-NO-ERRORS VALUE 'N'.
    05 UCIDWSMT-ERROR-CODE PIC S9(4).
    05 UCIDWSMT-ERROR-PARA PIC X(04).
    05 UCIDWSMT-ERROR-PROG PIC X(08).
    05 UCIDWSMT-CALL-REQ PIC X(06).
      88 UCIDWSMT-REQ-CHANGE VALUE 'CHANGE'.
      88 UCIDWSMT-REQ-COMMIT VALUE 'COMMIT'.
    05 UCIDWSMT-RETURN-STAT PIC X(06).
      88 UCIDWSMT-STAT-CHG-AUTH VALUE 'CHGÄUT'.
      88 UCIDWSMT-STAT-NOCHG-NOID VALUE 'NOREC'.
      88 UCIDWSMT-STAT-NOCHG-DUPID VALUE 'DUPID'.
      88 UCIDWSMT-STAT-NOCHG-DUPSSN VALUE 'DUPSSN'.
      88 UCIDWSMT-STAT-COMMITTED VALUE 'COMMIT'.
      88 UCIDWSMT-STAT-UNCOMMIT-NOID VALUE 'NOID'.
      88 UCIDWSMT-STAT-SUCCESS-BKOT VALUE 'BACKOT'.
      88 UCIDWSMT-STAT-NO-BACK-NOID VALUE 'NOBACK'.
      88 UCIDWSMT-STAT-INVALID-REQ VALUE 'INVRQ'.
      88 UCIDWSMT-STAT-IID100-PROB VALUE 'SUBRTE'.
  05 UCIDWSMT-RET-STAT-UCIDWS10 PIC X(06).
  05 UCIDWSMT-PASSED-FIELDS.
    10 UCIDWSMT-PASSED-IID-PRIOR PIC X(09).
    10 UCIDWSMT-PASSED-IID-AFTER PIC X(09).
    10 UCIDWSMT-PASSED-SSN PIC X(09).
    10 UCIDWSMT-PASSED-BIRTHDATE PIC X(10).
    10 UCIDWSMT-PASSED-NAME-LAST PIC X(30).
    10 UCIDWSMT-PASSED-NAME-FIRST PIC X(30).
    10 UCIDWSMT-PASSED-NAME-MIDDLE PIC X(30).
    10 UCIDWSMT-PASSKD-STAT PIC X(01).
    10 UCIDWSMT-PASSED-SYSTEM-NAME PIC X(30).
    10 UCIDWSMT-PASSED-SYSTEM-IID PIC X(09).
    10 UCIDWSMT-PASSED-CHANGED-BY PIC X(08).
    10 UCIDWSMT-PASSED-NAME-SUFFIX PIC X(04).
  05 UCIDWSMT-RETURNED-TIMESTAMP.
    10 UCIDWSMT-RET-TIMESTMP-OLDID PIC X(26).
    10 UCIDWSMT-RET-TIMESTMP-NEWID PIC X(26).
  05 FILLER PIC X(1762).
Any program calling UCIIDNUM must use the following structure to pass information to and from routine UCIIDNUM. This is an EXTERNAL structure. (Copylib member UCIDWSNO).

01 UCIDWSNO-INTERFACE EXTERNAL.
   05 UCIDWSNO-DIGITS-REQUESTED   PICTURE 9(01).
   05 UCIDWSNO-RANDOM-NUMBER      PICTURE 9(09).

Any program calling UCIIDCHK must use the following structure to pass information to and from routine UCIIDCHK. This is an EXTERNAL structure. (Copylib member UCIDWSCH).

01 UCIDWSCH-INTERFACE EXTERNAL.
   05 UCIDWSCH-NUM-WITHOUT-CHK-DGT PICTURE 9(06).
   05 UCIDWSCH-CHK-DIGIT-APPENDED  PICTURE 9(07).
"Random" ID Number Generation

This document indicates that in some circumstances (e.g., a newly hired employee) a new employee ID number be generated. This is referred to as a "random" number throughout this document. The assignment of ID numbers should be evenly distributed over the range of possible numbers. In addition, the ID number generated should not match any existing ID number on the IDB. The general procedure for generating a random ID number involves using the LE/370 function CEERANO. This function returns a random number which is formatted into a nine digit number.
Guidelines for Campuses Not Using Random ID Numbers

Campuses choosing not to use random ID number assignment as described in this document should modify UCIDNUM to reflect the campus' number assignment procedures. In cases where sequentially assigned numbers are used, UCIDNUM will probably need to access a file of assigned numbers to select the next number for assignment.

Campuses using SSN for ID number should modify UCIDNUM as well to supply SSN as the ID number.
Check Digit Algorithm

The purpose of the check digit is to reduce the probability of using an employee number where digits have been transposed, or one or more digits have been entered erroneously. While the check digit does not eliminate these types of errors, it will identify a majority.

Calculation of the check digit is as follows:

1. Multiply the digits of the six-digit employee number alternately by "1" and "2."

2. Add all the resulting digits together (include each digit of any two-digit products of step 1.)

3. If the low order digit of this sum is zero, then the check digit is zero. Otherwise, subtract the low order digit from 10 to derive the check digit.

EXAMPLES:

1. 6-digit ID: 4 1 7 2 3 6
   Multiplier: 1 2 1 2 1 2
   Products: 4 2 7 4 3 12
   Sum of digits: 23
   Check digit: 10 - 3 = 7
   7-digit ID: 4 1 7 2 3 6 7

2. 6-digit ID: 8 5 7 9 4 6
   Multiplier: 1 2 1 2 1 2
   Products: 8 10 7 18 4 12
   Sum of digits: 32
   Check digit: 10 - 2 = 8
   7-digit ID: 8 5 7 9 4 6 8

3. 6-digit ID: 5 8 6 9 5 4
   Multiplier: 1 2 1 2 1 2
   Products: 5 16 6 18 5 8
   Sum of digits: 40
   Check digit: 0
   7-digit ID: 5 8 6 9 5 4 0
Partial Match Condition

Throughout this document there is reference to a "partial match" condition when searching the IDB. The need to access records which are "close" matches to the record being searched for arises from the requirement that an individual be assigned a single identifier throughout their entire association with the University. In a rehire situation, the previously assigned record should be located even if the individual has changed their name or if a simple mistake was made in establishing one of the basic identifiers (i.e., SSN, birthdate) on the IDB.

Since the source of most errors in numeric fields is either transposition or the miskeying of a digit, the search for a partial match involves the full match of one of the basic identifiers and a partial match of the other.

When a search is made of the IDB for a match of target identifiers (name, SSN, birthdate), a positive match is made when a row matching both the SSN and birthdate are found in the IDB. The single record is returned.

If a positive match is not made, then the rows described below are returned as partial matches:

1. Any row where the name matches the target name.
2. Any row where the SSN matches the target SSN.
3. Any row where the birthdate matches the target birthdate and seven digits of the SSN are positional matches with the target SSN. (Positional match: Position n of the row SSN = Position n of the target SSN.)
6.0 Miscellaneous Changes

6.1 UCWMMNU

The Online Applications System Main Menu, UCWMMNU, will be modified to display the new IID Subsystem main menu selection.

6.2 One-Time IDB Loading

A one-time program will be developed to populate the IDB with initial data taken from the EDB. This will be an optional process since campuses may have another source for this data other than the EDB. If the EDB is not used as the initial source for the IDB data, then the IXB database must be populated with cross-reference data from the EDB.