PeopleSoft
Design Standards

Introduction

Any organization implementing the PeopleSoft Software must define its own naming and documentation standards for personalizing and adding objects/programs to the delivered system. Documentation of and adherence to naming conventions and documenting techniques will result in smoother upgrades by providing a clear means of distinguishing customer enhancements from the delivered PeopleSoft application objects and programs.

The following is a list of mandatory standards that must be adhered to when making, changing or creating new PeopleSoft objects and programs. PeopleSoft objects include Records, Fields, PeopleCode, Panels, and Menus. Additional Standards are given for SQR programs, Crystal, and Queries.

I. Record and Field Standards

PeopleSoft uses some naming conventions for records and fields.

Suffixes used on record/table names are as follows:

_VW A record definition that is physically implemented by defining an SQL view, such as, JRNL_LN_NPST_VW.

_DVW A record definition that is implemented by defining a dynamic SQL view (this kind of view is not a physical table), such as, JRNL_LN_DVW.

_TMP or **_WRK** A record definition that is used as a temporary work or derived table.

_TBL A record definition that contains data that controls the application as opposed to being maintained by the application. A "reference" table or "general" table.

_SRCH A view that is used as to select specific employee records to be displayed to the user via a panel. In addition, _SRCH views can be used to add department level security to SQR reports—if required by UHS.

Suffixes used on data elements are as follows:

AMT Amount; numeric value of currency type.

CD Code; user defined value from the Translate or other table.

CNT Count; Numeric value containing a count.

DT Date; format YYYY-MM-DD.

DTTM DateTime; format YYYY-MM-DD and HH:MM:SS

ID Identification; e.g. Operator ID

PCT Percent; stored as a decimal, i.e. 50% = .5

RT Rate; a numeric field expressing an amount per some unit

TM Time; hours, minutes, and seconds.

?? New records and fields MUST begin with the **UHS_** designator to distinguish them from PS-delivered fields and records. For example, a Flightfield for Salary Grade is added to the Employment record. The new field might be called UHS_SAL_GRADE.
?? NEVER delete a PS delivered field from a record.
? AVOID changing PS delivered field lengths where possible.
? NEVER rename PS delivered fields on a record. Instead, change the Short and Long Names. The new name will appear on any panel where the field is displayed.
? Exercise care when designating fields as LIST BOX items; especially, for high-volume tables such as PS_JOB. This will affect existing search views and indices. It will result in increased storage space and performance degradation.
?? Whenever it is applicable add new fields to the end of a record definition.
? AVOID use of the LONG EDIT field. Using this type of field increases the difficulty in ALTERing tables.
? Ensure consistent edits against prompt tables.
? Defining a field as "required" to PeopleSoft means that a character field can never be set to a space and a numeric field can never have a value of zero. Therefore, a value must be provided.
? Remember that the search record dictates the contents of a given dialog box.
?? Do not make Effdt a list box item.
? NEVER add derived fields to the PeopleSoft derived records:
DERIVED,
DERIVED_XXXX
where XXXX is a PeopleSoft functional extension (i.e. BEN, HR, PAY)

Translate Values

?? Never delete PS delivered translate values. Make them inactive if not needed.
?? If adding values and an XLAT edit check to an existing field, determine if the field exists elsewhere and if an XLAT edit check is needed on that record too.
?? Document new edits or new translate values in Record Descriptive Text under Record Properties in the Application Designer.
?? Only use the XLATTABLE (Translate Table, a generic valid values table for non-key fields) for non-key fields which meet the following criteria:

- Character type field whose assigned values need to be validated, and
- Length is 1 to 4 characters, and
- The field does not require more than a long/short description, code value, and effective date.

Fields with associated values on the XLATTABLE will sometimes appear as radio buttons on a panel. Please use radio button sparingly based on ease of usability for data entry.
?? Xlattable values should be of a static nature. They should not be values that will require constant modification.
?? If number of translate values are more than 100 consider putting data in a table.
? Any YES/NO or TRUE/FALSE fields should be specified as such in the Application Designer under Record Field Properties/Edits. Do not use the XLATTABLE for such fields.

Views

? Panel fields based on views should be display only.
? Any new view should be prefixed with **UHS_** and should be suffixed with standard PeopleSoft identifiers, _VW or _SRCH or _DVW.
? Associated table and view key fields MUST be consistent.
? If a field on a view is designated as an alternate key, ensure that it is designated as an alternate key on the associated data table as well.
? Understand the response time implications of the view.

Documentation

?? Document detail of changes within the descriptive text of the record. This should include adding fields to tables, and where these fields are being used or any other information that might be helpful. Note any panels these fields are added to. (This will be important information to have at the time of system upgrades.)
?? Changes within the Record Descriptive Text under Change Record Use should contain a control number (functional area and number combination), author, UHS, date, and brief description. For example:

BEN001, Author Name, UHS, MM/DD/YYYY
Added field Action_Dt to record

Please work with the DBA on the creation of all new views that enroll large tables or more than a few table joins.

II. PeopleSoft Panel Standards

?? Panel development should be done with a PC in the VGA mode (640 x 480 resolution). The Super VGA mode displays more screen space making it appear as if additional screen space is available for use. Use of this additional screen space would create problems when users access the system in VGA mode.

?? Follow the Object Naming conventions when creating new panels.

?? When beginning any modifications to an existing panel, a backup of the original panel should be made. You can accomplish this by opening the original panel and using the "save as" feature. The backup copy's name should be the original panel name with a suffix, but never overwriting existing versions. For example, if you were to make a change to Personal_Data1 panel first save it as Personal_Data1_00 (if it doesn't already exist) before making changes to the Personal_Data1 panel. Avoid renaming a PS delivered panel because PeopleCode could reference that panel.

? Try not to break up delivered PeopleSoft panel groups.

?? New panels should be prefixed with **UHS_**.

?? Major redesigns of delivered PS panels should be cloned and named with the appropriate UHS naming standard. Depending on your application you may choose to use a version number. ie: **UHS_XXXXX_01**.

? NEVER delete a field from a panel. Instead, make the field INVISIBLE.

? Key fields displayed on a panel should ALWAYS be display only.

? Use check boxes for YES/NO fields.

? Use edit boxes for character fields, number fields, and date fields.

? Use radio buttons only for character fields with XLATTABLE values that are static; that is, for fields whose possible values will NEVER change once established. If translate values are changed for a field which are represented by radio buttons on panels, these panels will require modification to account for the added, changed, or deleted XLATTABLE values.

? Up to 3 scroll bar levels can be specified on a panel. (memory)

? Each scroll bar should be associated with a specific record within a given parent-child hierarchy.

? Field order as defined under "View...Order" determines the flow of the cursor from field to field on the screen. Use this facility to set the order in which you would like the cursor to move.

? Ensure that all fields associated with a specific scroll bar are under that scroll bar in the Order.

III. PeopleCode Standards

On-line processing of data within PeopleSoft is controlled and managed by the Application Processor. PeopleCode programs are used to interact with this Application Processor to enhance and better control processing within a given panel or panel group. PeopleCode allows you to do such things as set default values on a panel upon panel entry, execute SQL statements to update the database, or conditionally change a panel based upon your own criteria.

PeopleCode programs are stored on the Data Designer and are associated with a specific field. For any field on a record, you may specify FieldDefault, FieldEdit, FieldChange, FieldFormula, RowInit, RowInsert, RowDelete, RowSelect, SaveEdit, SavePreChg, SavePostChg, SearchInit, SearchSave, and Workflow type programs. Each type of PeopleCode program executes at a different time within the processing flow of a panel or panel group. For more detailed information about PeopleCode types and programming, consult PeopleSoft's *PeopleCode Guide*.

UHS may require unique records/panels for which additional PeopleCode programs will need to be written. In order to differentiate changed and new programs from the delivered programs, a comment standard (see below) must be used when any existing PeopleCode programs are changed or new programs are written. For future upgrades, this will facilitate differentiating FlightPeopleCode from the vanilla PeopleCode that was delivered with the new release. Each time a developer changes PeopleCode or creates new PeopleCode programs, this comment standard MUST be used. Reference the bold type in the following example.

/* UHS - MM/DD/YYYY – *NAME (LAN ID)* */
(*UHS must be in uppercase letters and included in EVERY PeopleCode modification.)*
/* Description of the change being made - as many */
/* lines of description as needed. Please reference */
/* the PS Fix # or Customization # . */

New line(s)of code

/* END */

Additional PeopleCode Standards include:

- ?? Do not change vanilla code; rather, comment out the original and add new lines with the change(s). This allows for a developer to see exactly what was changed from the original.
- ?? Check the value of the %panel or %panelgroup global variables to control execution of PeopleCode programs on various panels.
- ?? If one field from a record (except for Related Display fields like Descrs) appears on a panel, all of the Row Level PeopleCode programs will execute.
- ?? Restrict the use of field level PeopleCode as it hinders on-line performance.

- ?? Any messages required through PeopleCode should not reference existing PeopleSoft Message Set numbers. ***Use Message Set numbers between 20000 and 29000 to add custom messages to the Message Set Catalog under the Utilities Menu. Individual messages within range can be sequentially assigned.***
- ?? To cut PeopleCode within the PeopleCode editor, use CTRL-X or use the Cut command button.
- ?? To copy PeopleCode within the PeopleCode editor, use CTRL-C or use the Copy command button.
- ?? To paste PeopleCode within the PeopleCode editor, use CTRL-V or use the Paste command button.
- ?? Determine whether anyone is making a change to the record you will be working on. If so, coordinate efforts.
- ?? When making PeopleCode changes, if PeopleCode does not work, **don't leave for the day with an error**. Temporarily remark out code until you can start working on it again.
- ?? Document change within the PeopleCode per UHS Naming Conventions.
- ?? NEVER add new functions to the PeopleSoft function libraries:
  FUNCLIB,
  FUNCLIB_XXXX
  where XXXX is a PeopleSoft functional extension (i.e. BEN, HR, PAY)
- ?? Create new derived records and function libraries for NEW derived fields and functions, e.g.
  UHS_DERIVED
  UHS_FUNCLIB
- ?? Use functions--whenever possible—for code that is performed in more than one place to ensure consistent processing and ease of maintenance.

IV. PeopleSoft Menu Standards

? NEVER delete menu items from the delivered PeopleSoft menus. If you do not wish for users to have access to specific menus or menu items, use Operator Security to hide these items from operator ids or classes.

V. Structured Query ReportWriter (SQR) Standards

SQR is usually the reporting tool of choice for **complex high-volume, batch reports** such as the **Deduction Register** or Quarterly **Tax Report**.

Additionally, PeopleSoft delivers various listing reports which detail the contents of a given table such as the valid value for fields. A high percentage of the delivered listing reports are queries.

Note that SQR reports can be executed from within the PeopleSoft on-line environment via the Process Scheduler. Reports that are executed in this fashion can be designated to run from the client workstation or the database server. If the SQR report runs from the client, the output will be written to the "C:\TEMP" directory. On the database server, the output will be written to the "/out" or "/log" directory in accordance with the naming convention that is associated with the DataBase Server. The naming convention for the output file is as follows:

<SQR name>.LIS - for Reports
<SQR name>.DAT - for file extracts

For example, a user executes the SQR report PAY002 (Deduction Register report) and chooses to have the report run on the client. The output report file would be called "C:\TEMP\PAY002.LIS". The SQR log file would be called c:\TEMP\SQR.LOG.

Several hundred reports written in the SQR (Structured Query ReportWriter) language are delivered with the PeopleSoft system. To enhance these vanilla reports, it is necessary to change the SQR programs delivered with the PeopleSoft product. PeopleSoft customers will usually use the SQR utility to build additional reports, interface programs, and other batch programs.

In order to ease portability you should resist using platform-specific SQL functions/expressions, where possible.

Except where changes are minimal, new programs should be created rather than modifying delivered vanilla programs. When making modifications to vanilla reports you should add a suffix to the report Id (e.g., "Move UPAY004 uhs' to $ReportID"). This will provide some clue to users and developers that it is not pure vanilla, if they were unfamiliar with the report and are comparing it against the PeopleSoft documentation. Include a space between your suffix (lowercase "uhs" is recommended) and the report id. The SQR naming convention standards are:

PeopleSoft delivered SQRs:
- PAYnnn - Payroll related programs
- PERnnn - Personnel related programs
- BENnnn - Benefits related programs

PeopleSoft delivered SQRs that have been modified for use at UHS:
- UPAYnnn - Payroll related programs
- UPERnnn - Personnel related programs
- UBENnnn - Benefits related program
- Note: The nnn, number portion, of the filename would match the delivered PeopleSoft nnn.

Programs that have been written specifically to meet a need at UHS:

UPYLnnn - Payroll related SQRs
UPRSnnn - Personnel related SQRs
UBFTnnn - Benefits related SQRs

UPYLUnnn - Payroll related SQUs - update programs
UPRSUnnn - Personnel related SQUs - update programs
UBFTUnnn - Benefits related SQUs - update programs

UPYL@nnn - Payroll related SQRs that produce a mailing address file
UPRS@nnn - Personnel related SQRs that produce a mailing address file
UBFT@nnn - Benefits related SQRs that produce a mailing address file

By using a different naming convention for the UHS developed programs, there will not be any confusion as to the origin of program. In addition, we would not have to be concerned about PeopleSoft eventually using a number in its filename that matches an existing UHS developed program.

Track SQR's & SQU's: Shared Drive \USER\EXCEL\SQR_LIST.XLS

Menu items for custom interfaces should be placed under a new *"Interface"* menu whenever practical.
However, there may be times when it is practical to add a new SQR to an existing menu item.

SQR Comment Standard

The following are excerpts from an SQR program. Please note the comments in bold type that are expected when any changes are made to an SQR program.

!---!
! Report Name: GLALEDGS.SQR - General Ledger - Summary !
!---!
! Description: This program will read all entries on the Ledger
!
! table and summarize the entries by account. Each journal
!
! associated with the account is then reported into debits and
!
! credits. An end balance is then calculated for the report. !
!---!
! Input Files: PS_Ledger, PS_GL_Account_Tbl **!**
! Output Files: !
!---!
! Prompts: Business Unit, Ledger, Fiscal Year, Begin Period, End Period !
! (user input via program or process scheduler)
!

```
!-------------------------------------------------------------------------------------------------!
! Tables Referenced: Ledger, GL_Acct_Tbl, Jrnl_Header, Jrnl_Ln
        !
!-------------------------------------------------------------------------------------------------!
! Mod Date      Author              ModId       Purpose                                           !
! ------------------------------------------------------------------------------------------------!
! 01/19/1995    UHS-S. Doe          SD001       Changed descr to KP_descr55.
        !
! 02/02/1995    UHS-J. Smith        JS001       Changed column names for                !
!                                               chartfields.
        !
!                                               Changed column locations of Debit       !
!                                               and Balances.
        !
!-------------------------------------------------------------------------------------------------!
#include 'setenv.sqc'    ! Set Environment Procedure
#include 'glbegin.sqc'   ! Begin Report Procedure
#include 'setup02.sqc'   ! Printer and page-size initialization
!-----------------------------------------------------------------------------!
! Procedure: Report                                                           !
! Desc:      This is the main report procedure.  It is executed once                  !
!            for each business unit to be processed.  It is called            !
!            from Process-Main.                                               !
!-----------------------------------------------------------------------------!
begin-procedure Report

  do Set-Dates

  do Set-Account-Types

begin-SELECT

A.Account                   (+1,#Account_Col,7)
!Begin …Comments
B.UHS_Descr55               (0,+#Descr_Col,55)      ! SD001
A.Currency_Cd               (0,+#Ledger_Curr_Col,3) ! JS001
!End Modification

SUM(A.Posted_Total_Amt) &Posted_Total_Amt

  move &Posted_Total_Amt to #Ending_Balance

  do Process-Journals
```

```
  let #Beginning_Balance = #Ending_Balance - #Total_Activity

  if $Activity = 'Y'
     print ' '                    (+1,1)
     print &A.Account             (+1,#Account_Col,7)
!Begin … Comments
     print &B.KP_Descr55          (0,+#Descr_Col,55)
     print &A.Currency_Cd         (0,+#Ledger_Curr_Col,3)  ! JS001
!End  Modification
  end-if

  print 'Beginning Balance:'      (0,#Total_Heading)
  do Format-Number(#Beginning_Balance, $out, '9,999,999,999,999.99pr')
  print $out                     (0,#Balance_Col)

  print 'Total Activity:'         (+1,#Total_Heading)
  do Format-Number(#Total_Activity, $out, '9,999,999,999,999.99pr')
  print $out                     (0,#Balance_Col)

  print 'Ending Balance:'         (+1,#Total_Heading)
  do Format-Number(#Ending_Balance, $out, '9,999,999,999,999.99pr')
  print $out                     (0,#Balance_Col)

  move  0 to #Beginning_Balance
  move  0 to #Total_Activity
  move  0 to #Ending_Balance
  move 'N' to $Activity
  print '_'                     (+1,1,173) fill
  print ' '                      (+1,1)

FROM  PS_Ledger A, PS_GL_Account_Tbl B
WHERE   = $GL_Account_Tbl_SETID
  AND B.Account = A.Account
  AND B.EffDt =
     (SELECT MAX(EffDt)
      FROM   PS_Gl_Account_TBL
      WHERE  SetID = B.SetID
        AND  Account = B.Account
        AND  EffDt <= $End_Date
        AND  Eff_Status = 'A')
  AND A.Business_Unit = $Business_Unit
  AND A.Ledger        = $Ledger
  AND ( ( A.Accounting_Period between 1 and #Select_End_Period and
         B.Account_Type in ([$Revenue_Expense_Accounts]) )
```

```
                OR
       ( A.Accounting_Period between 0 and #Select_End_Period and
         B.Account_Type in ([$Balance_Forward_Accounts]) ) )
  AND A.Fiscal_Year   = #Select_Fiscal_Year
  AND A.Account       like $Select_Account
  AND A.Statistics_Code = ' '
  AND A.Currency_Cd = $Select_Currency
!Begin … Comments
GROUP BY A.Account,B.UHS_Descr55, A.Currency_Cd  ! SD001
!End …Modification

end-SELECT

if $prcs_process_instance <> ''
  let $prcs_run_status = 'S'
  let $prcs_message_parm1 = 'Successful Completion'
  do Update-Prcs-Run-Status
end-if

end-procedure Report
```

UHS ***SQR Additional Standards:***

- All SQR reports must go through the change control process before they are allowed to be run in production (i.e. User Request ✍ FlightReport Administration ✍ Development ✍ Integration Testing ✍ QA ✍ Production.)

- Never change a program that exists in the directory PS delivered \sqr directory . This directory should stay untouched and vanilla.

- The development path for custom and modified PeopleSoft delivered SQRs is as follows:

Development/Unit Testing	- P:\SA760DMO\sqr
System Testing	- P:\SA760DMO\sqr
User Acceptance	- P:\SA760DMO\sqr
Production	- P:\SA760DMO\sqr

- If modifying a PeopleSoft delivered SQR, move a copy of the SQR from the PS delivered \sqr directory to a development \sqr directory.

- Use version control for the SQR's. A new SQR, version should be '00'. However, when modifying an existing SQR, the version should begin with '01'.

```
#Define version '-01'                    ! mod 1234    [place with other defines after Heading
    info]
concat {Version} with $Report ID         ! mod 1234    [place after ReportID variable
    initialization]
```

- If the user is to enter an input and/or output file, do not hard code the path. Have the user enter drive and path as part of the input/output file name.

- Separate procedures and sections with a commented line of asterisks.

- End every source code program with a line of asterisks that contains the phrase 'END OF REPORT'. This will aid in identifying any problems with file transfers that can often truncate files unexpectedly or add erroneous characters at the end of the code.

- SQR programs must always be API aware in order to properly update the Process Scheduler.

- Include running the SQR through SQRW as part of the testing.

- These same procedures apply to SQC modifications.

- The UH System SQR standard heading (UHSHDG00.SQC), standard body (UHSBDY00.SQC) and standard trailer modules (UHTRLR00.SQC) will be stored in the Y:\Project Team \ Technical Team \ Standards directory.

UHS Public Report Naming Conventions

- ***Report Id*** will have the following naming conventions:
  - For new reports:
    dddnnn
    where
    ddd=UHS
    Examples: UHS001, UHS002, UHS003
  - For modified PeopleSoft supplied reports:
    daaaaaa
    where
    d = The letter 'L' for UHS or some other identifier
    aaaaaa = PeopleSoft Report ID
    Examples: LPER705, LPER011, LBEN010

- ***Report Name*** will have the following naming conventions:
  - For new reports:
    dddnnn - xxx...xxx
    where
    dddnnn = is the report id name described above

xxx...xxx = business name
Examples: UHS001 - Division Table Report
UHS002 - Notice of Personnel Actions

? ? For modified PeopleSoft supplied reports use the PeopleSoft report name

?? The report file name corresponds to the *Report ID* .

?? A document is needed to track *Report IDs*. This document will be maintained by the reporting group.

VI. PS/Query, Excel & Crystal Standards

PS/Query

Public PSQuery Naming Conventions

Query Name will have the following naming conventions:

?? For new queries not tied to a Crystal report:
dddnnn - business name
where
ddd = UHS
Examples: UHS013-Missing Home Phone Number

?? For new queries tied to a Crystal report:
dddnnn - business name
where dddnnn = the Crystal Report ID
ddd = UHS
Examples: UHS003--Division Table Report

?? For modified PeopleSoft supplied queries:
daaaaaa - xxx...xxx
where
d = The letter 'L' for UHS or some other identifier
aaaaaa = PeopleSoft query prefix
xxx...xxx = PeopleSoft supplied query business name
Examples: NPER705-Location Table

PS/Query Hints

- Limit the number of records (i.e. joins) included in a single query. The more records, the slower the performance. Use the EMPLOYEES, BEN_PLAN_DATA, and BEN_PER_DATA tables. The EMPLOYEES table is populated by running an SQR program named PER099. The table contains a snapshot of employees' personal/demographic, employment and job/salary data as of the date indicated when the SQR program is launched. The BEN_PLAN_DATA and the BEN_PER_DATA tables are populated by running an SQR program named BEN100. The BEN_PLAN_DATA table is a snapshot of employee and dependent benefit plan data. The BEN_PER_DATA table is a snapshot of personal and dependent data. Note: **PER099 and BEN100 processes must be scheduled to run on a daily basis**.

- Use the *Prompt* option in the *Query Criteria* panel to specify a constraint or group of constraints at query run time instead of creating duplicate queries where a constant is varied. For example, prompt for *State* instead of creating duplicate queries for California, Hawaii, Washington, etc.

- Except where changes are minimal, new programs should be created rather than modifying PS delivered vanilla programs. New programs will be placed under a new UHS Custom Reports Menu.

- Queries should generally be saved as 'public' to enable use, study, and assistance by others and to avoid confusion later caused by duplicate versions.

Crystal

Public Report Naming Conventions

PeopleSoft delivered Crystal Reports:
PAYnnn - Payroll related reports
PERnnn - Personnel related reports
BENnnn - Benefits related reports

PeopleSoft delivered Crystal Reports that have been modified for use at UHS:
UPAYnnn - Payroll related reports
UPERnnn - Personnel related reports
UBENnnn - Benefits related reports
Note: The nnn, number portion, of the filename would match the delivered PeopleSoft nnn.

Crystal Reports that have been created specifically to meet a need at UHS:
UPYLnnn - Payroll related reports
UPRSnnn - Personnel related reports
UBFTnnn - Benefits related reports

By using a different naming convention for the UHS developed programs, there will not be any confusion as to the origin of program. In addition, we would not have to be concerned about PeopleSoft eventually using a number in its filename that matches an existing UHS developed program.

Track SQR's, SQU's and Crystal Reports: Shared Drive \USER\EXCEL\REPORTS_LIST.XLS

Menu items for custom interfaces should be placed under a new *"Interface"* menu whenever practical.
However, there may be times when it is practical to add a new SQR to an existing menu item.

?? The report file name corresponds to the *Report ID* .

?? The development path for custom and modified PeopleSoft delivered Crystal Reports is as follows:

Development/Unit Testing c:\user\crw

Crystal Hints

?? When you modify the underlying PSQuery that drives a Crystal report, you must relink the query to the report.
1.. Save the modified query by selecting the *File/Save* option on the *Query Field* panel.
2.. Launch Crystal Reports from the *Query Field* panel.
3.. Select *Database/Set Location* from the Crystal Reports main panel menu bar to display the *Set Location* pop up panel.
4.. Press the *Set Location* button to display the *Choose SQL Table* pop up panel.
5.. Select the query from the *SQL Tables* list and press *OK* to return to the *Set Location* panel.
6.. Press the *Done* button on the *Set Location* panel.
7.. Another pop up panel will appear asking whether you would like to modify the report. Reply *Yes* and modify the report to incorporate the query changes.

?? Except where changes are minimal, new programs should be created rather than modifying delivered vanilla programs.

?? Where possible, new reports should be compiled and icons created for the desktop rather than using a 'drop-down' report-bar (Process Scheduler) item. This will facilitate development, processing performance, and future maintenance.

?? New programs that need to be run via the Process Scheduler should be placed under a new UHS Custom Reports Menu.

Crystal Header/Trailer Standard

?? For letter size landscape, use P:\SA760DMO\crw\UHS_rptstd_landscape
?? For letter size portrait, use P:\SA760DMO\crw\UHS_rptstd_portrait
?? For legal size landscape, use P:\SA760DMO\crw\UHS_rptstd_legal

VII. SQL Query Standards

Only the Database Administrator is allowed to run SQL and DataMover(DMS) scripts in PRODUCTION environment. In addition, all migrations to PRODUCTION will be completed by the DBA (This is in compliance with the Change Control process).

VIII. COBOL Change Standards

It is strongly recommended that the delivered COBOL source code not be modified. Given that recommendation, sometimes it is necessary to make minor changes that do not affect the logic flow of the vanilla COBOL code. In some cases, the size of an array in a program must be increased or a new field must be added in the working storage section of a Copy book. This may be necessary due to the uniqueness of the data. Therefore, COBOL change standards must be implemented.

Whenever a change is required, the following rules should be incorporated to track the customized code for future maintenance.

When an object is identified for change it will be migrated from the \hr700\src\cbl\base library into a user customized library. A user customized directory could be called \hr700\src\cbl\UHS_custom.
Objects being changed should have the following comments added at the beginning of each object. Copy book objects are identified in source library as objects starting with “P*C”. Any object that has a “C” as the third character is identified as a PeopleSoft Copy book. Furthermore, objects that have a “P” as the third character is identified as PeopleSoft source code. All COBOL objects have a “cbl” suffix.

```
* -------------------User Modification Log ---------------------- *
* Project     Developer/Date/Revision/Description                  *
* --------    --------- ---- -------- ------------------------------ *
*                                                                    *
* PROJECT CODE -                                                    *
*          04/23/98 Version 1.00                                    *
*                                                                    *
```

```
    * ------------------------------------------------------------- *
```

Changes when applied must be incorporated into a marked block. Each block has a "START" and an "END". A block can use the following rule:

```
  **PPPPPP****START*****MM/DD/YYYY*************************************
V1.00         WHEN PS-RECORD-NM-JOB OF  W-WK                    UHSMOD I
V1.00                                        UHSMOD I
V1.00           PERFORM XL700-MOVE-JOB-TABLE                    UHSMOD I
     **PPPPPP****END******MM/DD/YY*************************************
```

Where "PPPPPP" is the beginning project when doing "START" and ending project when doing "END" in which the change was applied.
Where MM/DD/YYYY is the date on which the change was applied. On a "START" line it will indicate the date of the first change. On the "END" statement it will indicate the last time the block was modified.
Positions 1-6 are used to indicate a version associated with this change. Usually the maintenance log indicates a version/revision associated with a change.
Positions 73 - 80 are used to indicate the project in which the change is being applied. More than one project can fall within a marked block. Purpose of the marked block is to keep UH System customizations together.
PeopleSoft code is never physically removed from an object even when it no longer is applicable. The following shows how PeopleSoft code is marked and no longer has an impact on object processing. The original PeopleSoft code is removed by using "*$$" in positions 7 - 9. When removing a paragraph name, the paragraph information should be shifted by 3 bytes.

```
   **PPPPP****START****05/29/1998***************************************
V1.00*$$ 02 PROGRESS-REPORT       PIC S9(7)   VALUE 1     COMP-3.  PPPPP D
V1.00    02 PROGRESS-REPORT       PIC S9(7)   VALUE  +2000 COMP-3. PPPPP I
   **PPPPP****END******05/29/1998***************************************
```

IX. Logo Standard for Crystal Reports

??Use P:\SA760DMO\crw\UHS_sys_seal.gif