

# UNCLASSIFIED

eJMAPS Production Support Staff  
Standard Operating Procedures  
PeopleSoft Development Protocols

## 1. Statement of Purpose

Within development teams of any size, it is important that all personnel follow uniform protocols to keep the system synchronized and running smoothly. The following provides guidance on naming projects and objects, basic coding requirements, and basic testing requirements.

## 2. Naming Conventions

The naming convention for projects are as follows:

**EJ\_FUNCTIONAL\_AREA\_INITIALS\_YYYYMMDD**. For instance, my project for the inter COCOM transfer logic is named: **EJ\_INTR\_COCOM\_XFR\_PVH\_20100330**.

The naming convention for all other Application Designer objects is as follows:

**EJ\_MEANINGFUL\_OBJECT\_NAME**. Coming up with meaningful object names can be difficult due to the length limitations for object name fields. Just do the best you can and remember to preface the name with “**EJ**”. (Note: One exception to this rule is for custom copies of the **JOB DATA** component. It is better to name those components **JOB\_DATA\_XXX** without the EJ prefix. Otherwise they will not be included in the JOB DATA specific code references that identify those components by the string “JOB\_DATA” as the first eight characters of the name. If they are not included in the delivered JOB DATA code, they may not function properly.)

## 3. Commenting Conventions – Non Peoplecode Objects

Every new, or customized, definition must include an appropriate comment as part of its properties. If you are creating a new definition, you must enter a description of the new definition in the *Description* edit box. (If you use *File/Save As* to create a new object, remember to delete the existing *Description* and enter a new one.) Whether you are creating a new definition or modifying an existing definition, you must enter an appropriate comment in the *Definition* long edit box. The comment should include the name of the project, your initials or userid, the date, and a description of the modification. New comments should always be entered at the TOP of the *Definition* box. This is because the first line of the *Definition* box is used as the DESCRIPTION for the definition on the corresponding tool table.

## 4. Commenting Conventions – PeopleCode Objects

Each new or customized PeopleCode program must include a comment at the top of the program. New comments should follow the **existing** comments. The comment should

## UNCLASSIFIED

eJMAPS Production Support Staff  
Standard Operating Procedures  
PeopleSoft Development Protocols

include the name of the project, your initials or userid, the date, and a description of the modification.

### 5. Basic Coding Requirements

Most of our projects will include PeopleCode development, i.e., either new programs or changes to existing programs. Here are some PeopleSoft development guidelines to help you with this process.

- a) When selecting a PeopleCode event in which to place your code, the first thing you must consider is the firing sequence. You must choose the appropriate event in order to maintain the correct flow of your business logic. You can refer to the *PeopleBooks PeopleCode Developer's Guide > PeopleCode and the Component Processor* for an in-depth, and surprisingly easy to understand, explanation of the event firing sequence.
- b) **Errors and warnings** can only be coded in PeopleCode events that are error tolerant. There are only four error tolerant PeopleCode events: *Search Save, Row Delete, Field Edit, and Save Edit*. If you place an error or warning in any other event you may compromise the behavior and stability of your application. You may also cause an unrecoverable component error. This is becoming more of an issue as we roll out additional Component Interfaces and Messages.
- c) **Do NOT** use the **FieldFormula** event unless the field is on a **DERIVED** record.
- d) Use the object based syntax as often as possible. For instance, use **&PosnDataRowset.ActiveRowCount** instead of **ActiveRowCount(RECORD.POSITION\_DATA)**. This is important because Peoplecode is moving toward a true Object Oriented language and many of the obsolete built-in functions are being deprecated. In addition, some of the new coding features are more conducive to the object based PeopleCode, e.g., establishing the object "context" for component and/or page events.
- e) Always reference **DERIVED** fields using object based syntax. For instance use **&PosnPriority = &PosnDataRowset(&i).EJ\_DERIVED.EJ\_PRIORITY.Value** instead of **&PosnPriority = EJ\_DERIVED.EJ\_PRIORITY**. By design, derived fields are referentially ambiguous. This means the component processor only recognizes each derived field instance when referenced through the rowset, row and record.
- f) Use deferred processing whenever appropriate. Deferred processing suspends the execution of a PeopleCode program until the component is saved. This keeps the screen from flickering and reduces the number of trips to the server. The ability to use deferred processing depends on user requirements and may not be possible in every situation. When deferred processing is in use at the page/field level,

## UNCLASSIFIED

eJMAPS Production Support Staff  
Standard Operating Procedures  
PeopleSoft Development Protocols

*FieldEdit* and *FieldChange* events will not fire at the point where a field is changed. Keep this in mind as you are debugging your code.

- g) Remember to follow the rules for valid pages. Any field that is neither derived nor part of the primary record for the scroll must be defined as a *Related Display*. *Related Display* fields must follow the *Display Control* field in the page layout order. The primary record for a scroll is determined by the first non-derived record referenced in the scroll's page layout order. *Radio Buttons* must be grouped together. Whether or not they are grouped together is determined by their order in the page layout.
- h) When building new components, be sure to assign the correct access modes. The four access modes are *ADD*, *UPDATE DISPLAY*, *UPDATE DISPLAY ALL* and *CORRECT HISTORY*. Specific user access will be determined based on permission lists.
- i) Every component must be assigned a Search Record. The *Search Record* assigned to the component definition may be overridden by a *Search Record* assigned to the component's menu item. When this happens, it is the *Search Record* defined on the menu item that will be invoked at run time.
- j) If an existing record definition needs to be modified in such a way that the SQL definition must be rebuilt, e.g., a new field was added, remember to use the table *ALTER* option. If you attempt to *CREATE* the table instead of *ALTERING* it, you may end up blowing the table's contents away.
- k) Use the *FIND DEFINITION REFERENCES* search before modifying a definition with which you are not familiar. This is particularly important when working with field and record definitions. In addition, keep in mind that record and view modifications may also impact SQRs and queries.

Be sure to include ALL definitions in your project prior to migration. Set your project options to *When definition is modified, saved or deleted* to ensure that your definitions are inserted into the project as you work on them in the Application Designer. You must use the *UPGRADE* tab to insert definitions not defined in the Application Designer into the project. Non Application Designer definitions include Xlat Values, Process Definitions, Message Sets, Queries, Roles, Permission Lists, etc.

### 6. Basic Testing Requirements

You cannot consider your code UAT ready until you have validated it against as many of the following scenarios as applicable.

- a) Updating existing data
- b) Row Insert

## **UNCLASSIFIED**

### eJMAPS Production Support Staff Standard Operating Procedures PeopleSoft Development Protocols

- c) Row Delete
- d) Add
- e) Save
- f) Tests against both the IA and Permanent Party processes
- g) Tests for conditions being met as well as conditions not being met
- h) Tests for existing functionality that was not modified – regression testing
- i) Mass Load processes
- j) Reports
- k) Queries

The project developer should coordinate with another PSS team member to complete testing. Testing should include the steps documented in the migration form, as well as any obvious opportunities to simulate users performing inappropriate or illogical actions.