



PeopleTools v8.5x Integration Application Engine Course

Agenda

➤ Introductions

➤ About this Course

➤ Course Lessons

- Day 1: Technology Overview and Development Approach
- Day 2: Building Set-Based AE Programs
- Day 3: Advanced Topics and Troubleshooting

➤ Additional Q&A

- Throughout sessions, and at the end of each day

Introductions

Course Instructor – Douglas Pace



- 17 Years in Technology
 - Senior Developer
 - Development Manager
 - Usability Testing
 - User Interface Design

- 13 PeopleSoft Implementations
 - Conversion Lead
 - Technical Lead (Upgrades and Implementations)
 - Senior Developer

- Developer of Pace-Trace
 - Tracing Utility for PeopleSoft

About this Course

- Overview of the AE development process
- A “hands-on” course, with activities accompanying each area
- Interactive and Dynamically paced

AE Overview

- **PeopleSoft's Preferred Batch Language**
- **Replacing COBOL programs**
- **Database-Driven***
- **Typically Set-Based SQL Logic**
- **Supports PeopleCode Events**

Structure of an AE Program

➤ Made up of Sections, Steps, Actions

➤ Sections:

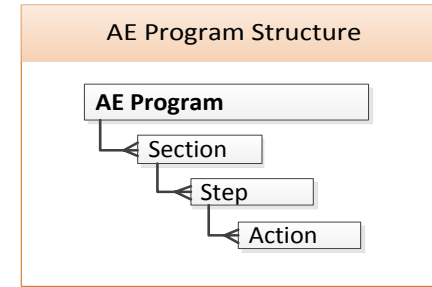
- All programs start with the MAIN Section
- Other Sections must be called

➤ Steps:

- Steps occur in sequence within a section
- Each step can have multiple actions

➤ Actions

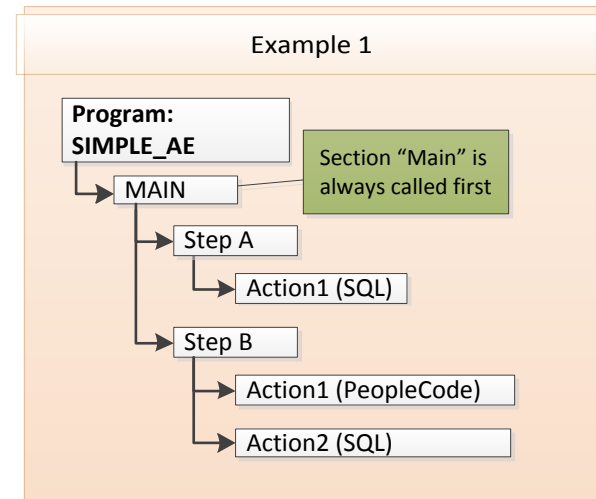
- The “meat” of an AE
- Where all the work occurs (PeopleCode, SQL, etc.)



Example AE Program Structures

➤ In Example 1, only a single section MAIN exists, with two steps

- Step A has a single SQL Action
- Step B has two actions



Example AE #2: Call Section

- Most AE Programs call sections from MAIN
- Notice the two steps under MAIN each call a section

The screenshot displays the configuration of an Autonomous Event (AE) program named 'MAIN'. The program is configured with two steps: 'INIT' and 'Process'. Each step is configured to call a specific section. A callout window shows the execution flow: Step 1: INIT calls 'Init.GBL.default.1900-01-01', which then calls 'Step 1: Step01', which in turn calls 'SQL'. Step 2: Process calls 'Process.GBL.default.1900-01-01', which then calls 'Step 1: InsDtIs', which in turn calls 'SQL'.

Program	Section Name	Program ID	Dynamic
MAIN	INIT	(current)	<input type="checkbox"/>
Process	Process	(current)	<input type="checkbox"/>

Section	Section Name	Program ID	Dynamic
Init	Step01	(current)	<input type="checkbox"/>
Process	SQL	(current)	<input type="checkbox"/>

Working with AE Programs in App Designer

➤ Reviewing overall program flow

- Bouncing between the two views
- The split-screen detail vs. the SQL edit window

➤ Adding App Engine programs to your project

- How parts are broken down (Sections, Steps) within the project items
- How this impacts migrations (orphaned code)

➤ Finding Code in Existing Programs

- Identifying programs called by this program (and so on)

➤ Additional Topics for Later

- Resolving Meta-SQL

State Records

- Work record during the execution of your program
- Variables which persist throughout execution
- Standard naming convention to use AET as record suffix (eg. PO_ST_AET)
- Can be SQL Tables or Work Record
 - State recs stored as SQL Tables must be keyed by process instance
 - Use SQL tables if using commit-points to maintain values in state rec after commit
 - If SQL table, include column PROCESS_INSTANCE
- Placing Values into State Variables

– Using SQL:

```
%Select(PRCSPURGE_AET.SERVERNAME)
SELECT SERVERNAME
FROM PSSERVERSTAT
WHERE SERVERNAME <> %Bind(PRCSPURGE_AET.SERVERNAMERUN)
AND SERVERSTATUS = '3'
AND ( %DateTimeDiff(LASTUPDDTTM, %CurrentDateTimeIn) < 10)
```

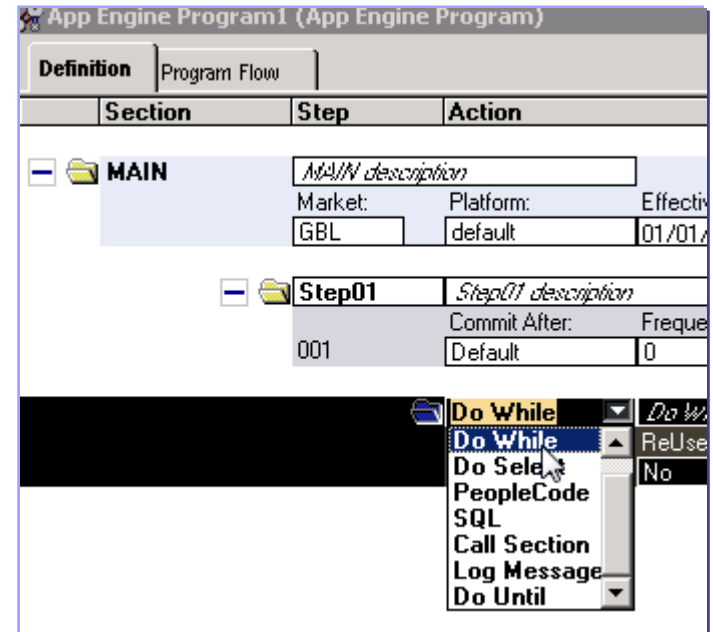
– Using PeopleCode:

```
ZPV_RUN_GEN_AET.SETID = "SHARE";
```

Actions (one step can have several):

➤ Possible types of Actions for a Step

- Do Actions
 - Do While
 - Do When
 - Do Select
 - Do Until
- SQL (Update/Delete/Insert/Select)
- PeopleCode
- Log Message
- Call Section



➤ One Step can have more than one action, but only one of each type

- For example, you can have a SQL and PeopleCode action, but only one of each.
- If you have 2 SQL statements to run, you will need two Steps, with each having an associated SQL Action.

Do Actions: Do When

- Run the associated SQL and perform the statements beneath if at least one row is returned
- Requires selecting to a state variable using %Select or %SelectInit

```
%SelectInit(DUMMY_FIELD)
SELECT 'X' FROM PS_INSTALLATION
WHERE EXISTS
  ( SELECT 'X' FROM %Table(IN_FL_ER2_TAO)
    WHERE PROCESS_INSTANCE =
%Bind(PROCESS_INSTANCE)
AND MESSAGE_NBR = 94)
AND %Bind(SYNCH_FLAG) = 'N'
```