PeopleSoft PeopleTools 8.54 Application Engine
Developed developers, analysts and system admins, PeopleSoft PeopleTools v8.53/8.54 Application Engine training quickly introduces students to the PeopleSoft application development leveraging The Application Engine tool. Through scenarios, real-world experiences from implementers, and hands-on activities, students learn how to develop new and modify existing PeopleSoft Application Engine programs.

**Prerequisites:**
- Basic Programming knowledge OR PeopleCode knowledge
- PeopleTools knowledge

### Technology Overview and Development Approach

- **Technology Overview**
  - Database Driven, Batch Programs, Set-Based
  - Structure of AE (Sections, Steps, Actions)
- **Working with AE Programs in App Designer**
  - Viewing program structure
  - Adding your AE Program to your Project
  - Migrating AE Programs
  - Finding code in existing programs
- **AE Action Types**
  - Do Actions: While, When, Select, Until
  - Understanding Action Execution Order
  - SQL
  - PeopleCode
  - Log Message
  - Call Section

### Technology Overview and Development Approach (cont'd)

- **Example 1: Creating an AE Shell**
- **State (AET) Records**
  - Work Record throughout execution
  - Retrieving and Storing Values in State Records
- **Creating Process Run Pages**
  - Process Run Pages Overview
  - Designing Run Control Records (OPRID, RUN_CNTL_ID)
  - Creating Run Control Pages (PRCSRUNCNTL_SBP)
  - Building the Component (PRCSRUNCNTL_SBP)
  - Connecting a Process to a Component (Process Definition)
Building Set-Based AE Programs

- Temporary Tables and AE
  - Understanding Temp Table Architecture: How TAO Table instances are Used for isolating set Processing
  - Structure of TAO Tables (PROCESS_INSTANCE as key)
  - Table Instances and Where They Are Set (Temp Table tab for each AE Program), online instance count in PeopleTools Options
  - Online vs. Batch Instances
  - Referring to table instances [%Table()]
  - Temporary Table Usage (PeopleTools > Application Engine > Review Temp Table Usage)

- Inactivating Steps
  - How to mark steps inactive
  - Finding all steps marked inactive in a program (SQL)

- Writing SQL for AE (Using Meta-SQL)
  - Used to interact with state variables
  - Simplifies SQL Coding Substantially
  - %InsertSelect: To insert into one table by selecting from another
  - %Select: To take results and place into state variables

Building Set-Based AE Programs (cont'd)

- Writing PeopleCode for AE
  - Take care to avoid row-level PeopleCode
  - How to access state values in PeopleCode
  - Affecting AE Program Execution with PeopleCode (Exit(0))

Troubleshooting

- Running an AE from the Command-Line
- Building Restartable AE Programs
  - Why Restartability is Necessary
  - How Restart Works (Commits and PS_AERUNCONTROL)
  - Forcing Programs to Start from Beginning
- Dynamic Section Calls
  - Using state variables to store the section name
  - Other ways of conditionally calling sections (Do When)
  - Best when there are many potential logic paths (Sections to call)
Troubleshooting

- Using the App Engine Request Page
  - PeopleTools > Application Engine > Request AE
  - Specifying Bind Variables through Run Page
- Starting AE From PeopleCode
  - Initiating App Engine Programs from PeopleCode (CallAppEngine)
- Using AE Libraries
  - Similar to the App Class or FUNCLIBs
- Dynamic SQL in PeopleCode
  - Pulling in text from a SQL object using %SQL(SQLID)
  - Putting in a where condition via %BIND({statevar},NOQUOTES)
- Troubleshooting AE
  - Finding within all code of an AE
  - ToolsTable Overview
  - Finding other programs called by this AE (tools SQL)
  - Finding inactive steps
  - Techniques to Handling Abends
  - Reviewing Trace timings

Troubleshooting

- Tracing in AE Programs
  - Turning on Tracing (Process Definition)
  - Setting Trace Flags (-Trace, -ToolsTraceSQL, -ToolsTracePC)
  - Reviewing the Trace File
  - AET vs. Trace SQL files