

1.2.3 Application Engine

PeopleSoft Application Engine enables you to develop, test, and run Application Engine programs. You use PeopleSoft Application Engine to develop batch or online programs that perform high-volume, background processing against your data. In PeopleSoft Application Engine, a program is a set of SQL statements, PeopleCode, and program control actions that enable looping and conditional logic. You define PeopleSoft Application Engine programs in PeopleSoft Application Designer.

1.2.3.1 Environment and Architecture

Application Engine programs are stored in Tools tables, along with the SQL they execute.

The screenshot shows the 'Open Definition' dialog box. The 'Definition' dropdown is set to 'App Engine Program'. The 'Selection Criteria' section includes:

- Name: AEMASSCHNG
- Project: All Projects
- Description: (empty)
- Owner ID: All Owners
- Program Type: All Programs

 The 'Definitions matching selection criteria:' table is displayed below, with the following data:

Name	Description	Type	Program Type
AECLEANUP	AE Abend Clean up	Program	Standard
AEDAEMONMGR	Daemon Manager Program	Program	Standard
AEMASSCHNG		Program	Standard
AEMINITEST		Program	Standard
AE_DETYA_AET	DETYA Rpt generation process	Program	Standard
AE_DETYA_FIL	DETYA Rpt generation process	Program	Standard
AE_SYNCIDGEN		Program	Standard

The program consists of Actions, within Steps, within sections. The Actions invoke a SQL statement, invoke PeopleCode, or call another section:

The screenshot shows the 'Definition' window in PeopleTools, displaying a program flow for 'Execute a Mass Change'. The flow is organized into sections and steps:

- MAIN** (Execute a Mass Change):
 - DriveMC** (Extract and run the MC SQL):
 - 001: Commit After: After Step, Frequency: 0. Action: `%Select (AEMASSCHNG_AET.MC_STMNT_SEQ, AEMASSCHNG_AET.MC_SEQ) SELECT MC_STMNT_SEQ, MC_SEQ FROM PS_MC_DEFN_SQL WHERE MC_DEFN_ID = %Bind(AEMASSCHNG_AET.MC_DEFN_ID) ORDER BY MC_STMNT_SEQ, MC_SEQ`
 - Do Select** (Get each MC SQL): ReUse Statement: No
 - PeopleCode** (Initialize the SQL and...): On Return: Abort
 - Call Section** (Build and run the SQL): Section Name: RunSQL, Program ID: AEMASSCHNG, Dynamic:
- RunSQL** (Build and run the MC SQL):
 - BuildSQL** (Build the SQL statement):
 - 001: Commit After: Default, Frequency: 0, On Error: Abort, Active:
 - Do Select** (Get each SQL chunk): ReUse Statement: No, Do Select Type: Select/Fetch

Two callouts provide details for the DriveMC step:

- DriveMC (aestep)**: `AEMASSCHNG_AET.SQLTEXT = "";`
- SQL Query**: `%Select (AEMASSCHNG_AET.MC_STMNT_SEQ, AEMASSCHNG_AET.MC_SEQ) SELECT MC_STMNT_SEQ, MC_SEQ FROM PS_MC_DEFN_SQL WHERE MC_DEFN_ID = %Bind(AEMASSCHNG_AET.MC_DEFN_ID) ORDER BY MC_STMNT_SEQ, MC_SEQ`

Actions may include looping logic (Do Select) based on the fetching of multiple rows.

1.2.3.2 Execution

The code, logic and SQL are stored in PeopleTools tables such as PSAESECTDEFN, PSAESTMTDEFN and PSSQLTEXTDEFN.

At run time, the code, logic and SQL are retrieved from these tables, cached on disk and in memory, and executed by the driver program "psae.exe".

SQL is passed through to the I/O routine 'PSSYS.dll', and then to a platform specific conversion routine 'PSODBC.dll'. Here, ODBC calls are passed through DB2 Connect to be executed in DB2. When running on NT or UNIX, the SQL comes in through DDF. When running under USS, the calls come in through RRSF (Resource Recovery Services Access Facility)

Application Engine programs can be executed at the command prompt or using shell scripts:

```
user01/hr890/bin/psae.exe -CT DB2UNIX -CS -CD HR890 -CO PS -CP * -R ADHOC -AI AEMINITEST -I 1 -OT 2 -OF 13 -OP /user01/temp/
```

```

export PS_HOME=/u/data006/pt810bh
export PS_DB=DB2
export PS_PLT=OS390_2_8
export PS_SERVER_CFG=/u/data006/pt810bh/pt810bh.cfg
export STEPLIB="CEE.SCEERUN:HOLROYD.PT810.USS.RUNLIB:HOLROYD.PT810.RUNLIB:GSK.SKLOAD:DSN610.SDSNLOAD:$STEPLIB"
export LIBPATH=$LIBPATH:/usr/lib:$PS_HOME/bin
export PATH=$PATH:$PS_HOME/bin:$PS_HOME/bin:/usr/local/bin
DSNAOINI="/u/data006/pt810bh/dsnaoini"
/u/data006/pt810bh/bin/psae -CT DB2ODBC -CD PT810BH -CO PTADMIN -CP PTADMIN -R 0
-AI AETESTPROG -IO -DR -TRACE 3 -TOOLSTRACESQL 31

```

Alternatively, AE can use JCL submitted from z/OS using a couple of utility programs

```

//BPXCOPY EXEC PGM=BPXCOPY,
// PARM='ELEMENT(AETEST1) TYPE(TEXT) PATHMODE(0,7,7,7)'
//SYSUT1 DD DSN=HOLROYD.PT810.AETEST,DISP=SHR
//SYSUT2 DD PATH='/u/data006/pt810bh'
//SYSTSPRT DD SYSOUT=*
//*****
//OEBATCH EXEC PGM=BPXBATCH,REGION=0M,
// PARM='SH /u/data006/pt810bh/AETEST1'
//STEPLIB DD DSN=CEE.SCEERUN,DISP=SHR
//STDOUT DD PATH='/u/data006/pt810bh/aetest.out',
// PATHOPTS=(OWRONLY,OCREAT,OTRUNC),PATHMODE=SIRWXU
//STDERR DD PATH='/u/data006/pt810bh/aetest.err',
// PATHOPTS=(OWRONLY,OCREAT,OTRUNC),PATHMODE=SIRWXU
//*
//STDENU DD DSN=HOLROYD.PT810.AETEST,DISP=SHR

```

BPXCOPY copies a member containing Environment variable settings and the psae command (AETEST1) over into an HFS file (/u/data006/pt810bh/AETEST1) in USS.

BPXBATCH then picks up this file and executes it as a shell script.

Application Engine programs can call Cobol programs. These are normal Cobol programs, but they access DB2 using the Call Attach Facility, not the TSO Attach Facility. So they have been compiled and linked with DSNALI as opposed to DSNELI, exist in a separate load library, and run using a different plan.