Keys to Success for Progress Application Server for OpenEdge (PASOE) in Production

Roy Ellis
ellis@progress.com
• Understanding PASOE
• Run the newest versions
• Configure PASOE for your application
• Monitoring PASOE
• Memory
• Summary
Understanding PASOE

PASOE is _NOT_ the Classic AppServer
Classic AppServer
Progress Application Server for OpenEdge
GTO

- Car
- Transport people
- Uses Gas
- No Computer
- 4 Speed Transmission
- 10 miles to the gallon

Tesla

- Car
- Transport people
- Uses Electricity
- Is a Computer
- No Transmission
- 110 miles to gallon
Classic AppServer
- Runs ABL code
- Supports:
  - REST
  - WSA
  - WebSpeed
  - AppServer
  - Mobile

PASOE
- Runs ABL code
- Supports:
  - REST
  - SOAP
  - WEB
  - APSV
  - Mobile
Classic AppServer

- No Web Server
- 1 Agent = 1 ABL Session
- No Built-in Security
- Adapters for REST, WSA, AIA
- WebSpeed messenger
- AppServer protocol
- Requires AdminServer
- Load Balancing via NameServer

PASOE

- It is a Web Server!
- 1 Agent = many ABL Sessions
- Spring Security
- REST, SOAP, APSV Built-in
- WEB Transport Built-in
- New Web Handler
- Stand Alone
- Load Balancing using industry standard products
Understanding PASOE

Architecture
Architecture: Components

Classic AppServer Components

AppServer
  ↓
Agent
  ↓
NameServer
  ↓
AdminServer

PAS for OpenEdge Components

AppServer
  ↓
Agent
  ↓
NameServer
  ↓
AdminServer

Client
  ↓
Session Manager
  ↓
MSAgent
  ↓
ABL Sessions

APSV (AIA)
SOAP (WSA)
REST/Mobile
WebSpeed

© 2017 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
Architecture: Sample

Classic AppServer Components

- AdminServer
  - State-Aware
- AppServer
  - 50 Agents (1 ABL Session)

PAS for OpenEdge Components

- Client
- NameServer
  - State-Free
  - Rest/Mobile
- Session Manager
  - APSV (AIA)
  - SOAP (WSA)
  - REST/Mobile
  - WebSpeed
- MSAgent
  - 150 ABL Sessions
Architecture: Multi-Session Agent

- Multiple client requests at the same time
- Supports both Session-Managed and Session-Free request simultaneously
- Manages database shared memory connections
- Uses much less system resources
- Handles multiple times more transactions than the single session AppServer agents
# PROMON - RDBMS Self-service Connections

User Control: by user number for all tenants

<table>
<thead>
<tr>
<th>Usr:Ten</th>
<th>Name</th>
<th>Domain</th>
<th>Type</th>
<th>Wait</th>
<th>Table:Part</th>
<th>Dbkey</th>
<th>Trans</th>
<th>PID</th>
<th>Sem</th>
<th>Srv</th>
<th>Login Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>0</td>
<td>BROK</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8068</td>
<td>0</td>
<td>0</td>
<td>04/01/15 19:14</td>
</tr>
<tr>
<td>5</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASA</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9596</td>
<td>2</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>6</td>
<td>root</td>
<td>0</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>1412160</td>
<td>0</td>
<td>9596</td>
<td>3</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>7</td>
<td>root</td>
<td>0</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>924864</td>
<td>0</td>
<td>9596</td>
<td>3</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>8</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9596</td>
<td>4</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>9</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>030912</td>
<td>0</td>
<td>9596</td>
<td>4</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>10</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>1427776</td>
<td>0</td>
<td>9596</td>
<td>5</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>11</td>
<td>root</td>
<td>0</td>
<td>MON</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1359</td>
<td>5</td>
<td>0</td>
<td>04/02/15 15:10</td>
</tr>
</tbody>
</table>
## PROMON - RDBMS Self-service Connections

User Control: by user number for all tenants

<table>
<thead>
<tr>
<th>Usr:Tn</th>
<th>Name</th>
<th>Domain</th>
<th>Type</th>
<th>Wait</th>
<th>Table:Part</th>
<th>Dbkey</th>
<th>Trans</th>
<th>PID</th>
<th>Sem</th>
<th>Srv</th>
<th>Login Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>0</td>
<td>BROK</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8068</td>
<td>0</td>
<td>0</td>
<td>04/01/15 19:14</td>
</tr>
<tr>
<td>5</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASA</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9596</td>
<td>2</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>6</td>
<td>root</td>
<td>0</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>1412160</td>
<td>0</td>
<td>9596</td>
<td>3</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>7</td>
<td>root</td>
<td>0</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>924864</td>
<td>0</td>
<td>9596</td>
<td>3</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>8</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9596</td>
<td>4</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>9</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>030912</td>
<td>0</td>
<td>9596</td>
<td>4</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>10</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>1427776</td>
<td>0</td>
<td>9596</td>
<td>5</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>11</td>
<td>root</td>
<td>0</td>
<td>MON</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1359</td>
<td>5</td>
<td>0</td>
<td>04/02/15 15:10</td>
</tr>
</tbody>
</table>
# PROMON - RDBMS Self-service Connections

User Control: by user number for all tenants

<table>
<thead>
<tr>
<th>Usr:Ten</th>
<th>Name</th>
<th>Domain</th>
<th>Type</th>
<th>Wait</th>
<th>Table:Part</th>
<th>Dbkey</th>
<th>Trans</th>
<th>PID</th>
<th>Sem</th>
<th>Srv</th>
<th>Login Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>0</td>
<td>BROK</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8068</td>
<td>0</td>
<td>0</td>
<td>04/01/15 19:14</td>
</tr>
<tr>
<td>5</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASA</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9596</td>
<td>2</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>6</td>
<td>root</td>
<td>0</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>1412160</td>
<td>0</td>
<td>9596</td>
<td>3</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>7</td>
<td>root</td>
<td>0</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>924864</td>
<td>0</td>
<td>9596</td>
<td>3</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>8</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9596</td>
<td>4</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>9</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>030912</td>
<td>0</td>
<td>9596</td>
<td>4</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>10</td>
<td>root</td>
<td>-4</td>
<td>SELF/PASN</td>
<td>--</td>
<td>6</td>
<td>1427776</td>
<td>0</td>
<td>9596</td>
<td>5</td>
<td>0</td>
<td>04/01/15 19:19</td>
</tr>
<tr>
<td>11</td>
<td>root</td>
<td>0</td>
<td>MON</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1359</td>
<td>5</td>
<td>0</td>
<td>04/02/15 15:10</td>
</tr>
</tbody>
</table>
PASOE at Instance Create and Run-time

$DLC/servers/pasoe

PAS for OE (template)

lib
bin
*.sh
conf
webapps
common/lib
openedge
elextras

Copy & tailor
Full copy

(webapps [*.war])

$WRKDIR/oepas1

PAS for OE Instance

*.sh
conf
logs
temp
work
webapps
openedge

OS Process

PAS for OE Process

lib
bin
*.sh
conf
logs
temp
work
webapps
common/lib
openedge

Full copy

Copy & tailor

tcman create

(CATALINA_HOME)

(CATALINA_BASE)

tcman start
# PASOE Development versus Production

<table>
<thead>
<tr>
<th>PAS for OE Development</th>
<th>PAS for OE Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can compile code</td>
<td>Cannot compile code</td>
</tr>
<tr>
<td>Non-secure configuration</td>
<td>Secure configuration</td>
</tr>
<tr>
<td>Test server instance in $WRKDIR</td>
<td>No test server instances</td>
</tr>
<tr>
<td>Remote administration included</td>
<td>Remote administration optional</td>
</tr>
<tr>
<td>Tomcat remote admin enabled</td>
<td>Tomcat remote admin optional</td>
</tr>
<tr>
<td>OpenEdge remote admin enabled</td>
<td>OpenEdge remote admin optional</td>
</tr>
<tr>
<td>Built-in oeabl web application (ROOT)</td>
<td>Built-in oeabl web application (ROOT)</td>
</tr>
<tr>
<td>All transport deployed and enabled</td>
<td>All transports deployed but disabled</td>
</tr>
<tr>
<td>Restricted: 5 concurrent requests and 1 agent</td>
<td>Unrestricted: concurrent requests and agents</td>
</tr>
<tr>
<td>defaultHandler=</td>
<td>defaultHandler=</td>
</tr>
</tbody>
</table>
Run the newest versions
Run the newest versions

- 11.6.4 and 11.7.2
- Many new features (more on these later)
- Memory leak bugs fixed
- Soap Out (agent calls out to soap) bugs fixed
- SSL/HTTPS bugs fixed
  - OpenSSL updated
  - Other thread-safe bugs fixed
Configure PASOE for your application
The defaults are always wrong!

- Don’t go into production without testing your PASOE instance under expected load and modifying the configuration!
- We have since reset the defaults to reflect best practices
- Tomcat settings
  - `tcman config` will list all of the configurable values for Tomcat
  - `tcman config psc.as.executor.maxthreads=300` – Will set Tomcat incoming threads to 300 (the default by the way)
  - `psc.as.https.maxqueuesize=100` – thread overflow if executor maxthreads are full
The defaults are always wrong!

- **openedge.properties**
  - `maxAgents=10`
  - `minAgents=0` 0 is off, >1 maintain this number
  - `numInitialAgents=1`
  - `agentStartLimit=1` only start one agent at a time
  - `maxConnectionsPerAgent=16` max concurrent connects per agent
  - `maxABLSessionsPerAgent=200` max agent sessions *
The defaults are always wrong!

maxthreads=300
maxqueueSize=100

maxAgents=10
maxConnectionsPerAgent=16
Total 160

-n 100
Example configuration

maxthreads=200
maxqueuesize=100
maxAgents=2
maxConnectionsPerAgent=100
Total 202
Test, Test, Test

- Some tips for load testing, machine sizing, performance setting

1. Always run your tests with your driver on one machine, PASOE on a separate machine, and your database on a third machine

2. Start with low client load

3. What to monitor during performance testing
   1. Check round-trip time
   2. Check system resources (top - Load)
Test, Test, Test

2. What to monitor during performance testing (continued)
   1. Check PASOE process memory (java)
   2. Check MSAgent process memory (_mproapsv)
   3. Check MSAgent Session Memory (oemanager REST API, others)

3. When to upgrade machine
   1. When round-trip time starts to fall
   2. When CPU load matches CPU number
   3. When memory is exhausted (starts to swap)
Monitoring your production PASOE
Monitoring your production PASOE

- Jconsole
- OE Manager REST API
- Tomcat Manager
- JMX Query
- Log File Monitoring
Jconsole

- GUI based tool for accessing the Tomcat and PASOE JVM directly
- Direct access to all JMX beans and objects in the Tomcat/PASOE Session Manager
OE Manager REST API

- Must be installed to the PASOE instance
  - tcman deploy $DLC/servers/pasoe/extras/oemanager.war
  - Results in: ~/webapps/oemanager
- Can now access PASOE information via HTTP/REST calls
  - http://localhost:8810/oemanager/applications/oepas1/metrics

```json

{  
  result: {  
    requests: 0,  
    writeErrors: 0,  
    reads: 0,  
    concurrentConnectedClients: 0,  
    maxQueueDepth: 0,  
    writes: 0,  
    maxConcurrentClients: 0,  
    readErrors: 0,  
    reserveConnectionTimeouts: 0,  
    timesQueued: 0,  
    avgQueueDepth: 0,  
    type: "OE_BROKER",  
    startTime: "2017-11-08T08:05:01.733-05:00",  
    accessTime: "2017-11-08T08:33:34.913-05:00"
  },  
  operation: "GET SESSION-MGR METRICS",  
  versionNo: 1,  
  versionStr: "v11.6.4 (2017-09-23)",  
  errmsg: "",  
  outcome: "SUCCESS"
}
```
OE Manager REST API

- Available via HTTP/REST
- Must secure if using in Production
- Change the default passwords
  - ~/conf/tomcat-users.xml
- Limit access to the URL (Remote Address Filter)
  - webapps/oemanager/WEB-INF/web.xml
OE Manager REST API

- [http://localhost:8810/oemanager/applications/oepas1/metrics](http://localhost:8810/oemanager/applications/oepas1/metrics)
  - Like “asbman -query” from classic

- [http://localhost:8810/oemanager/applications/oepas1/sessions](http://localhost:8810/oemanager/applications/oepas1/sessions)
  - All current client sessions in the Session Manager

- [http://localhost:8810/oemanager/applications/oepas1/agents/22484/sessions](http://localhost:8810/oemanager/applications/oepas1/agents/22484/sessions)
  - Shows all agents ABL sessions, start and end times, Session memory

  - Shows all current requests in the agent
Tomcat Manager

- Web based management and monitoring
  - Tomcat process
  - ABL and Web Applications
- Data on current requests to the web server
- Data on memory size of the web server (including the Session Manager)
- More
JMX Query

- Script based access to the Tomcat JMX beans
  - Allows gathering information locally without opening ports
  - Like Jconsole, without GUI interface

- Available in 11.6.4 and 11.7.2

- ~/bin/oejmx.sh(.bat)
  - Self documented in the script/batch file
  - White paper to follow
Memory
Memory: MSAgent

- Memory usage will appear to be multiplied
- We use less memory in total, but its all in 1 executable
- If you classic AppServer agent routinely used 2 Gig of memory
- Moving to PASOE and setting the maxConnectionsPerAgent=100
- Results in an msagent somewhere under 200 (historically 140) Gig of memory
Memory: MSAgent

“Although PASOE is our new web application server, it is also a very good tool for exposing memory leaks!”

- Having 100 ABL sessions in a single agent (versus 1 session in Classic)
- Memory issues and leaks are multiplied by 100 times!
- This means a 100 Meg leak of 24 hours
- Is now 10 Gig over the same 24 hours in PASOE
Memory: MSAgent

- Progress has fixed some memory issues highlighted by PASOE
  - Client-Principal, SSL, Web Handler

- Tools to help you find memory leaks (new and improved)
  - New – memory-checker
    - Ability to dump memory allocation of object to a file per ABL session
    - Note: run with one agent and one session
  - Updated – Dynamic Object leakchecker.p
  - Dump a ABL session stack (like prostack) via the oemanager REST API
Memory Checker

DEF NEW GLOBAL SHARED VARIABLE iactcnt AS INT NO-UNDO.

iactcnt = iactcnt + 1.

IF iactcnt = 50 THEN
    _memory:START-MEM-CHECK(LEAK-DETECTION, TRUE, ?).
ELSE IF iactcnt = 200 THEN
    _memory:STOP-MEM-CHECK().
Memory leakchecker

- Turn on Dynamic Object logging
  - In the configuration file, openedge.properties
  - agentLogEntryTypes=ASPlumbing, DB.Connects, DynObjects.*
    - DynObjects.Class
    - DynObjects.DB
    - DynObjects.Other
    - DynObjects.XML
    - DynObjects.UI
Memory leakchecker

- Turn on Dynamic Object logging
- In an sessionActivateProc

```plaintext
DEF NEW GLOBAL SHARED VARIABLE iactcnt AS INT NO-UNDO.
DEF NEW GLOBAL SHARED VARIABLE clogentries AS CHARACTER NO-UNDO.

iactcnt = iactcnt + 1.

IF iactcnt = 50 THEN
  DO:
    ASSIGN clogentries = log-manager:log-entry-types
    log-manager:log-entry-types = log-manager:log-entry-types + ",DYNOBJECTS.*:4".
  END.
ELSE IF iactcnt = 200 THEN
  DO:
    log-manager:log-entry-types = clogentries.
  END.
```
Memory leakchecker

- Once the information is logged to the oepas1.agent.log
- Run the new leakchecker.p
- [https://knowledgebase.progress.com/articles/Article/P133306](https://knowledgebase.progress.com/articles/Article/P133306)
- This will show any leaking objects by showing any created but not deleted dynamic objects
ABL session stack

- This will dump ABL session information for an ABL session
- If the session is using large amounts of memory
- This will dump the current ABL stack for review
- [http://localhost:8810/oemanager/applications/oepas1/agents/22484/sessions/7/stacks](http://localhost:8810/oemanager/applications/oepas1/agents/22484/sessions/7/stacks)
ABL session stack

```json
{versionNo: 1, versionStr: "v11.6.4 ( 2017-09-23 )", errmsg: "", outcome: "SUCCESS", result: {
  ABLStacks: [{
    AgentSessionId: 7,
    StartupParams: "-pf C:\Progress\116\OpenEdge\startup.pf, -cpinternal ISO8859-1, -cpstream ISO8859-1, -cpcoll Basic, -cpcase Basic, -d mdy, -numsep 44, -numdec 46, (end .pf), -logginglevel 2, -logfile C:\OpenEdge\116\WRK\oepas1\logs/oepas1.agent.log, Propath: ", C:\OpenEdge\116\WRK\oepas1\openedge, C:\Progress\116\OpenEdge\tty, C:\Progress\116\OpenEdge\tty\ablunit.pl, C:\Progress\116\OpenEdge\tty\adecomm.pl, C:\Progress\116\OpenEdge\tty\adecomp.pl,... ",
    Databases: [],
    PersProcs: [{Proc: "C:\Progress\116\OpenEdge\tty\webutil\_wstyle.r", ProcId: 1079}, ...
    OO4GLObs: [{Class: "OpenEdge/Logging/Logger.r", ObjectId: "001051"}, ...
    Callstack: [],
    Status: "Idle"},
  },
  operation: "GET SESSION RUNTIME STACKS"
}
```
Memory: Reclaiming System Memory

- First try removing high memory Agent sessions
  - `curl -v -X DELETE -u tomcat:tomcat http(s)://host:port/oemanager/applications/agents/agentId/sessions/sessionId`

- If that doesn’t work you can “stop” an Agent
  - This will return memory to the system
  - This REST API allows 10 seconds for all currently running requests to finish before stopping
  - [https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/pasoe-admin%2Fstop-an-agent.html%23](https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/pasoe-admin%2Fstop-an-agent.html%23)

- Set `minAgents=(a number 2 or greater)`
  - This will restore agents to the pool when an agent is stopped (or crashes)
Memory: Reclaiming System Memory

- Surprise!
- oom_kill
  - Unix systems
  - Kills the largest memory using process when memory threatened
  - MSAgent was being killed!
Memory: Resource Timeout

- `agentListenerTimeout=300000`
- `agentWatchdogTimeout=3000`
- `connectionWaitTimeout=3000`
- `idleAgentTimeout=300000`
- `idleConnectionTimeout=300000`
- `idleResourceTimeout=0`
- `idleSessionTimeout=300000`
- `requestWaitTimeout=15000`
- `socketTimeout=3000`
- `sessionTimeout=180`

**These values are milliseconds**

- If running with a “backup” agent may want to disable this value (0)

**Must be set to >0 to enable, NO timeouts enabled when set to 0**
Memory: Tomcat

- OpenClient connections
  - Be sure to release and dispose of OpenClient connections
  - Previously dotNet and JAVA connect to the UBroker
    - UBroker would clean up abandoned connections
  - dotNet and JAVA now connecting to the PASOE web server
    - Web server leaves the connections open for another request
  - We have seen tomcat thread leaks because OpenClients are not closing their connections
  - https://knowledgebase.progress.com/articles/Article/NET-Open-Client-disconnect-messages-not-received-by-PASOE
Call to Action

- Make the time to understand PASOE
- Stay up-to-date with the newest version
- Test and properly configure PASOE
- Test and look for memory issues
- Set up monitoring for PASOE