PASOE VS Legacy AppServer
Consultingwerk Ltd.

- Independent IT consulting organization
- Focusing on OpenEdge and related technology
- Located in Cologne, Germany, subsidiary in UK
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- 27 years of Progress experience (V5 … OE11)
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration

http://www.consultingwerk.de/
Introduction

- Legacy is classic 😊
- Flexible and extensible part of a Service-oriented architecture
- Standards-based transaction engine
- Provides user interface independence
- Enables business logic to be easily distributed and reused
- Centralized business logic provides a single point to manage access to data
Agenda

- Overview legacy AppServer
  - Overview PASOE
  - Comparison
  - Setup and Configuration
  - Client connection from the ABL
  - Deployment Options / Strategies
Overview legacy AppServer

- AdminServer
- Client
- AppServer
- NameServer
- Agent
  - 1 ABL Session
- AIA WSA
- WebSpeed
- REST/Mobile
Overview legacy AppServer

- **AppServer Broker**
  - Manages pools of re-usable AppServer agents
  - Manages client connections and requests for AppServer agents to execute ABL

- **AppServer Agents**
  - Executes OpenEdge ABL procedures
  - The reuse of AppServer agents is determined by Operating mode
Overview legacy AppServer

- AdminServer
  - State-Aware
  - AppServer
    - 50 Agents
- NameServer
  - State-Free
  - Rest/Mobile
  - AppServer
    - 50 Agents
- Stateless
  - AppServer
    - 50 Agents
- Client
  - AIA WSA
    - WebSpeed
  - REST/Mobile

PASOE VS Legacy AppServer
Agenda

- Overview legacy AppServer
- **Overview PASOE**
  - Comparison
  - Setup and Configuration
  - Client connection from the ABL
  - Deployment Options / Strategies
Overview PASOE

- **PASOE VS Legacy AppServer**

## PAS for OpenEdge
- APSV (AIA)
- SOAP (WSA)
- REST/Mobile
- WebSpeed

## Session Manager

## MSAgent
- ABL Sessions
Overview PASOE

- Just one Multi-Session Agent able to utilize all kind of clients and operation modes
Overview PASOE

- Progress proprietary network protocol has been dropped
- Tomcat does not speak TCP
Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison
  - Setup and Configuration
  - Client connection from the ABL
  - Deployment Options / Strategies
Comparison

Legacy Appserver

- Must have an AdminServer
- AdminServer starts the NameServer and AppServer, then the AdminServer is just used to monitor and manage
- Client connects to NameServer (optional) and is then redirected to the AppServer
- If the Client needs to connect through the internet (HTTP/HTTPS) you must install and configure separate and distinct adapters AIA/WSA/REST/Mobile
Comparison

PASOE

- Start the PAS for OE instance
- Client connects using HTTP/HTTPS always
- AdminServer is not necessary (except for running OEMgmt on the same machine or publishing)
- There is NO NameServer (load balancing and fault tolerance are handled another way)
Comparison

- **Session Models** and **Operating Modes** for legacy AppServer
  - Specified when AppServer is configured
  - Determines how client requests are dispatched to individual Application Server Agents
  - Considerations at Design time
    - Application context
    - Performance optimization
Comparison

- Connection management
  - Session managed model
    - Client sends requests over „bound“ connection
      - Requests are handled sequentially
    - Operating Modes
      - State-reset (Progress V8.2)
      - State-aware (Progress V9)
      - Stateless (Progress V9)
Comparison

- Connection management
  - Session free model
    - Client sends request on any available connection
      - Requests handled in parallel
    - Operating Mode
      - State-free (OpenEdge 10.0A)
Comparison (Architecture, capabilities)

- **Operating Modes**
  - State-reset
    - One client per Application Server Agent
    - Session state is reset on disconnect
  - State-aware
    - One client per Application Server Agent
    - Session state is maintained across connections
  - Stateless
    - Connection managed by AppServer Broker
    - Many clients per application server agent
    - Context must be managed programmatically
Comparison

- AppServer operating modes can be programmatically adopted to PASOE
- Unbound by default
- Bound session can be achieved by
  - Having the client run a remote persistent procedure
  - Setting `SESSION:SERVER-CONNECTION-BOUND-REQUEST` in the Connect and Disconnect procedure’s
- Only ABL clients, .NET and Java OpenClient can use bound sessions
- Web, SOAP and REST clients cannot use this
Comparison

Migrating classic state-reset and state-aware operating mode
To achieve the same client **bound** and **reset** behavior in PASOE, include the following as the first statement in your Connect event procedure:

```
SESSION:SERVER-CONNECTION-BOUND-REQUEST = TRUE.
```

This will fully disable the execution of Activate and Deactivate event procedures for client requests over this connection until this attribute is set to FALSE, but leave them available for use by session-free client connections.
Comparison

Also include the following as the last statements before exiting your Disconnect event procedure:

```
SESSION:SERVER-CONNECTION-BOUND-REQUEST = FALSE.
QUIT. (only for state-reset)
```

The QUIT statement in PASOE effectively performs the same ABL session reset operation that occurs in the classic AppServer after the Disconnect event procedure executes.
Comparison

- Migrating classic stateless and state-free operating mode
- No changes are required to event procedures to migrate applications running on a classic stateless AppServer.
Comparison

- More details on migrating from legacy Appserver to PASOE
## Comparison

### Startup and Shutdown procedures

<table>
<thead>
<tr>
<th>Procedure name</th>
<th>Legacy AppServer</th>
<th>PASOE</th>
<th>PASOE procedure name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Executed on Server startup</td>
<td>agentStartupProc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Executed on Server shutdown</td>
<td>agentShutdownProc</td>
</tr>
<tr>
<td>srvrStartupProc</td>
<td>Executed on Session startup</td>
<td></td>
<td>sessionStartupProc</td>
</tr>
<tr>
<td>srvrShutdownProc</td>
<td>Executed on Session shutdown</td>
<td></td>
<td>sessionShutdownProc</td>
</tr>
<tr>
<td>srvrConnectProc</td>
<td>Executed on Client connect</td>
<td></td>
<td>sessionConnectProc</td>
</tr>
<tr>
<td>srvrDisconnProc</td>
<td>Executed on Client disconnect</td>
<td></td>
<td>sessionDisconnProc</td>
</tr>
<tr>
<td>srvrActivateProc</td>
<td>Executed on Server startup</td>
<td>Executed on Session activate</td>
<td>sessionActivateProc</td>
</tr>
<tr>
<td>srvrDeactivateProc</td>
<td>Executed on Server shutdown</td>
<td>Executed on Session deactivate</td>
<td>sessionDeactivateProc</td>
</tr>
</tbody>
</table>
Comparison

Full comparison of event procedures can be found here:
https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/pasoe-migrate-develop%2Fcomparing-event-procedures-between-the-appserver.html%23
Comparison

Improvements in PASOE

- Multi-Session Agent
  - Single process supports multiple, concurrent, ABL sessions
  - Sessions share resources
  - Manages shared memory connects

- Session Manager integrated with Tomcat
  - No separate Java processes for Ubroker, NameServer or AdminServer (optional)
  - Removes one network copy in stack
# Comparison

<table>
<thead>
<tr>
<th></th>
<th>Classic</th>
<th>PAS for OE</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scalability</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client connections</td>
<td>221</td>
<td>1312</td>
<td>493%</td>
</tr>
<tr>
<td><strong>Server Resources</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>10 CPUs</td>
<td>5.2 CPUs</td>
<td>192%</td>
</tr>
<tr>
<td>Memory</td>
<td>2.1 GB</td>
<td>670 MB</td>
<td>313%</td>
</tr>
<tr>
<td>Transactions</td>
<td>203 tps</td>
<td>1698 tps</td>
<td>736%</td>
</tr>
<tr>
<td><strong>Client performance</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenEdge</td>
<td>YMMV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebSpeed performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Round trip</td>
<td>YMMV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison

- **Setup and Configuration**
  - Client connection from the ABL
  - Deployment Options / Strategies
Setup and Configuration

Resources

- AdminServer (1 Item)
  - desktop-2lcvivc
- AppServer (6 Items)
  - desktop-2lcvivc.TestServer1
  - desktop-2lcvivc.asbroker1
  - desktop-2lcvivc.bpsbroker1
  - desktop-2lcvivc.esbbroker1
  - desktop-2lcvivc.icrepos
  - desktop-2lcvivc.restbroker1

Resource Summary

Resource Count: 0
Setup and Configuration

- Add new Resource from the Resources Toolbox
Setup and Configuration

- Add new AppServer Resource
- Name the Instance
- Save - the new Instance is created
- Configure the AppServer as needed
Setup and Configuration

- Broker configuration
  - Port number (default 3090) may already in use
  - Operating Mode
### Setup and Configuration

- **Agent configuration**
  - Startup parameters
  - PROPATH

<table>
<thead>
<tr>
<th>General</th>
<th>Logging Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server executable file: \bin\proapsv.exe</td>
<td>Server log filename: TestServer1\server.log</td>
</tr>
<tr>
<td>Server startup parameters: /db C:\Work\AppServer\Talk\DB\sports2000.db -H localhost</td>
<td>Basic</td>
</tr>
<tr>
<td>PROPATH:</td>
<td>Append to server log file: yes</td>
</tr>
<tr>
<td>Minimum port number: 2202</td>
<td>Server logging level: ASPlumbing.DB Connects</td>
</tr>
<tr>
<td>Maximum port number: 255</td>
<td>Server log file threshold size: 0</td>
</tr>
<tr>
<td>Flush statistical data:</td>
<td>Maximum number of server log files: 3</td>
</tr>
</tbody>
</table>

**Pool Range**

- Agent configuration
- Startup parameters
- PROPATH
Setup and Configuration

- **Agent configuration**
  - Number of Agents to start
  - Startup and Shutdown Procedures

Used f.e. to initialize the backend session

---

**Pool Range**

- Initial number of servers to start: 1
- Minimum servers: 1
- Maximum servers: 10

**Advanced Features**

- 4GL debugger enabled: Not set
- Activate procedure: Not set
- Deactivate procedure: Not set
- Connect procedure: Not set
- Disconnect procedure: Not set
- Startup procedure: Not set
- Shutdown procedure: Not set
- Startup procedure parameters: Not set
- Execution Time Limit: 0
Setup and Configuration

- Ubroker.properties

```
[UBroker.AS.TestServer1]
appserviceNamelist=TestServer1
brkrDebuggerKeyAlias=default_server
brokerLogFile=@{WorkPath}\TestServer1.broker.log
controllingNameServer=NS1
environment=TestServer1
initialSrvrInstance=1
keyAlias=default_server
mqBrokerLogFile=@{WorkPath}\TestServer1.mqbroker.log
mqServerLogFile=@{WorkPath}\TestServer1.mqserver.log
portNumber=3190
PROPATH=@{WinChar Startup}\PROPATH;@{WorkPath};c:\Work\AppServerTalk\ABL
srvrLogFile=@{WorkPath}\TestServer1.server.log
srvrStartupParam=-db C:\Work\AppServerTalk\DB\sports2000.db -H localhost
    uuid=83a1dda7d9d8c1fe:-47431f8:15fa1ff2833:-75fb
```
Setup and Configuration

- Add new Resource from the Resources Toolbox
Setup and Configuration

- Add new Progress Application Server
- Name the new Instance of PASOE
- Define the Instance Directory
- On Windows - define the Shutdown port
- Supply Tomcat manager the login and password details
Setup and Configuration

- Define Startup parameters
- Set PROPATH
- Set Session procedures
Setup and Configuration

- **Openedge.properties**

```plaintext
[AppServer]
checkMetrics=1
applications=PASTestServer
statusEnabled=1
allowRuntimeUpdates=0

[AppServer.SessMgr.PASTestServer]
agentLogEntryTypes=ASPumbing,DB,Connects
agentStartUpParam=-T "${catalina.base}/temp" -db C:\Work\AppServerTalk\DB\sports2000.db -H localhost
checkMetrics=3
agentLogFile=${catalina.base}/logs/PASTestServer.agent.log

[AppServer.Agent]
numInitialSessions=5
agentTableParam=-n 200
sessionStartupProcParam=
sessionConnectProc=
agentStartupProc=
noSessionCache=0
lockAllExtLibs=
agentMinPort=62002
sessionStartupProc=
usingThreadSafeExtLibs=
workDir=${CATALINA_BASE}/work
binaryUploadMaxSize=0
```
Setup and Configuration

- Openedge.properties

```java
[AppServer.Agent.PASTestServer]
numInitialSessions=2
uuid=http://DESKTOP-2LVICV:8443/PASTestServer
keyAliasPasswd=
PROPATH=c:\Work\AppServerTalk\ABL,${CATALINA_BASE}/webapps/ROOT/WEB-INF/openedge,${CATALINA_BASE}/openedge`
```
Setup and Configuration

- Another way of creating a new PASOE instance is using the `tcman` command:
  
  ```
  c:\progress\OpenEdge117_64\servers\pasoe\bin\tcman create –p 8830 –s 8831 –p 8832 –j 8833 TestPASOE
  ```

- The `tcman` command can also be used for the Startup and shutdown of a PASOE instance:
  
  ```
  {pasoe folder}\bin\tcman {start|stop}
  ```

- TRIM Agents (that were the old days 😊)

- On PDSOE you need to TRIM Session, not the whole MSAgent
Setup and Configuration

Classic AppServer tools
- asbman –query
- OpenEdge Explorer
- OpenEdge Management

PASOE
- Management REST API
  - The “oemanager” application must be installed for access to the REST API
- JMX access
  - Using Jconsole
  - Locally by PID
  - Remotely - “tcman.sh/.bat feature JMXLifecycle=on”
- Nagios
  - 3rd party tool recommended by Tomcat for monitoring
  - Uses the tools above to monitor, alert and graph
    - OpenEdge Explorer and Management
Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison
- Setup and Configuration
  - **Client connection from the ABL**
- Deployment Options / Strategies
Client connection from the ABL

- REST/Mobile clients: URL still uses ../rest/..
- SOAP clients: ../soap/.. instead of ../wsa/..
- WebSpeed: ../web/.. Instead of ../cgi-bin/cgi.sh/..
- OpenEdge clients URL connection format only
  - URL http://myhost:8810/(appname)/apsv
    No (appname) assumes ROOT -URL http://myhost:8810/apsv

Reference: Connecting to AppServers Using a URL

NOTE: Since all of the old web servlets are now encapsulated in PASOE we use these “transports” to identify the communication: rest, soap, apsv and web
Client connection from the ABL

- A SERVER object is used to connect from ABL source code to either type of AppServer
  
  ```abl
  DEFINE VARIABLE hServer AS HANDLE NO-UNDO.
  CREATE SERVER hServer.
  ```

- Connecting to legacy AppServer
  ```cmd
  hServer:CONNECT("-AppService TestServer1 -H localhost -s 3190").
  ```

- Connecting to PASOE
  ```cmd
  hServer:CONNECT("-URL http://localhost:8080/apsv").
  ```
Client connection from the ABL

- Demo
  - Get simple response
  - Get data from DB

- Show how to use the same sourcecode in a local session to connect to a legacy AppServer and PASOE
Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison
- Setup and Configuration
- Client connection from the ABL

- Deployment Options / Strategies
Deployment Options / Strategies

- Many roads lead to Rome 😊
- Options you may need
  - Port forwarding from a Gateway
  - Rewrite Rules (www.myDomain.com/{AppName})
  - LoadBalancing

- Never use PASOE Development as a production instance!!!
  - Not only because of licensing
<table>
<thead>
<tr>
<th>PASOE Development</th>
<th>PASOE 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>
</tbody>
</table>
Questions