

■ *Understanding PASOE Progress Application Server for OpenEdge*

Mike Fechner



Mike Fechner



35 years of experience in Progress
(from Version 5 to OpenEdge 12)

Active member of the OpenEdge
community and speaker at international
conferences

Expert in software architecture, object-
oriented design, and web technologies

Modernization in Focus



**Modernization of
Legacy OpenEdge
Applications**



**Deep Technical
Expertise**



**Global IT Partner
with Local
Presence**



**More than
Consulting – We
Deliver Tools &
Solutions**

An urgent reminder!

- OpenEdge 11.7 was retired **219 days** ago!
- Windows 10 retired October 14th 2025
- Windows 12 expected – according to rumors
- No security fixes for OpenEdge 11.7 after April 1st
- No bug fixes for OpenEdge 11.7 after April 1st
- No new platform certification after April 1st
- No formal Windows 12 support for OpenEdge 11.7
- ...



Progress Application Server for OpenEdge



OpenEdge Application Server use-cases

- Since OpenEdge 8.3: **distribution of load** from (GUI client-server clients)
 - Execution of Business Logic closer to the Database
 - Improve latency of Database requests
 - Asynchronous requests from client (OpenEdge 10)
- Backend for **web applications**, WebSpeed
- OpenClient to support **alternative UI technologies** as clients
 - Visual Basic / Active X
 - Java
 - .NET

OpenEdge Application Server use-cases

- Application **integration** via SOAP Web Services
- Backend for **web and mobile Applications** via REST Adapter (OpenEdge 11.2)
- Application integration via REST/RESTful Web Services (OpenEdge 11.2)

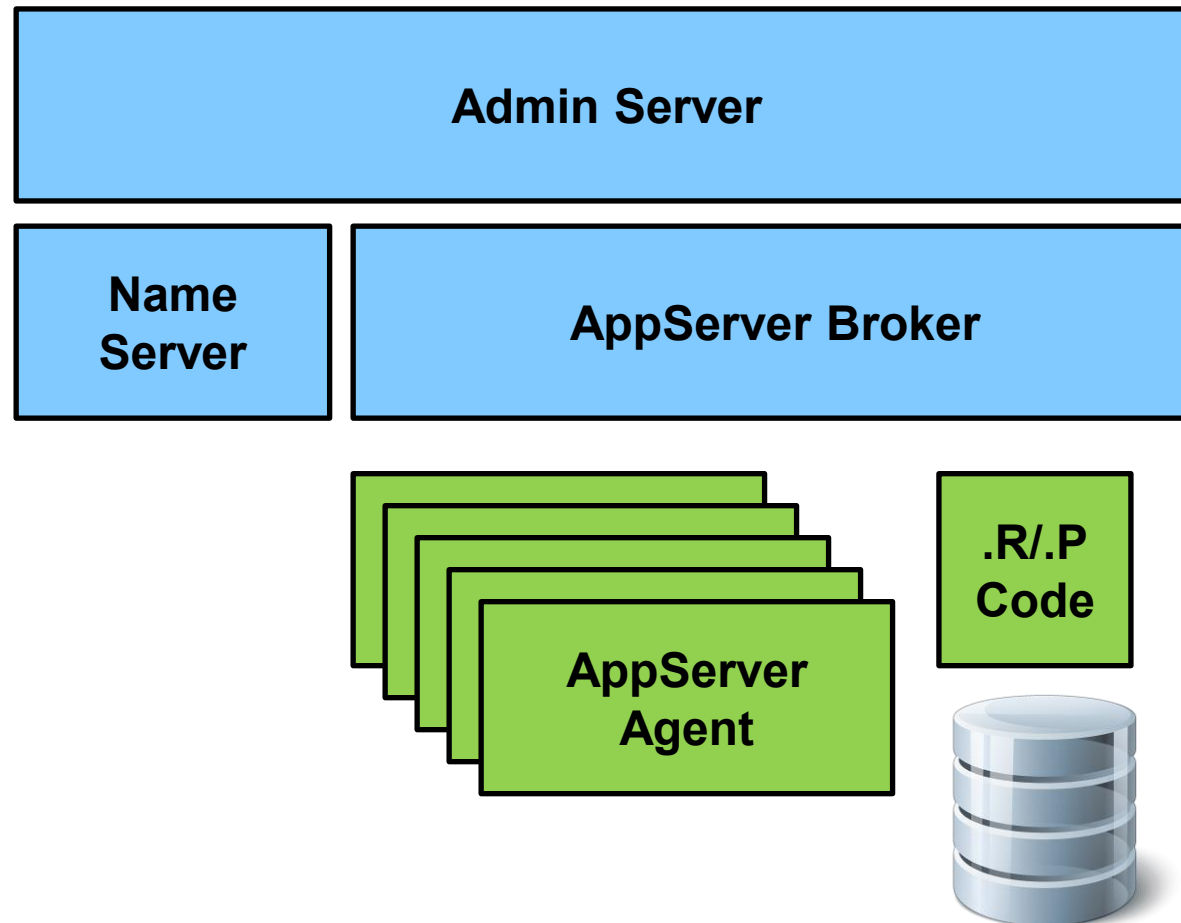
PASOE

- Progress Application Server for OpenEdge
- PASOE is the *new* AppServer for OpenEdge
- Introduced in OpenEdge 11.5 with APSV, REST and SOAP transport
- WEB transport added in OpenEdge 11.6
- Enhanced in OpenEdge 11.7
- Starting OpenEdge 12.0 the *only* AppServer for OpenEdge as the classic AppServer is retired

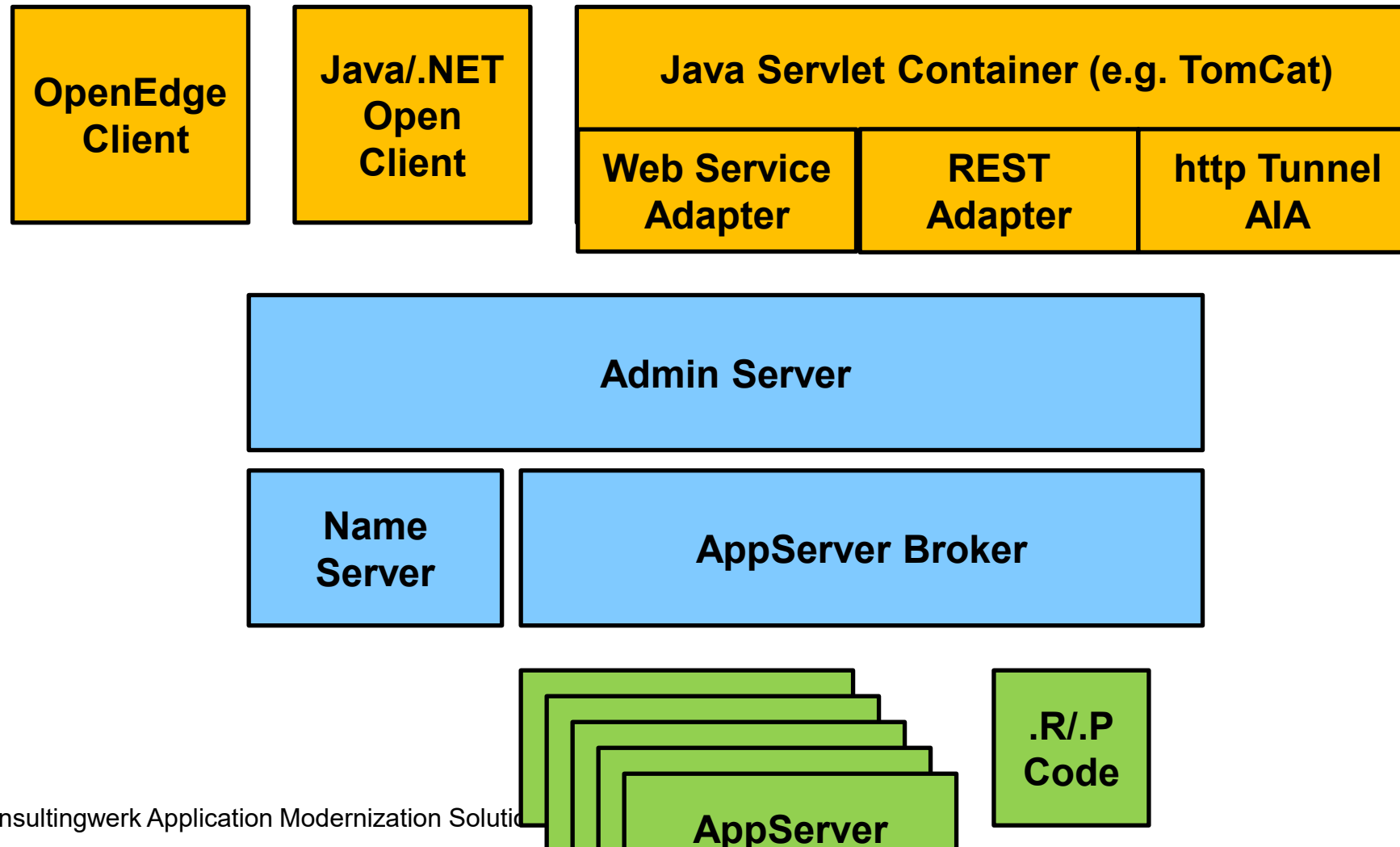
PASOE

- **64 bit only** for AppServer
- 32/64 bit for Clients (e.g. ABL GUI)
- Supported Docker images for development and production
- Build on top of Apache Tomcat – also to leverage wide set of authentication and authorization options from the Tomcat ecosystem
- No specific built-in load-balancing or fail-over solution
- **No dependency on Admin-Server framework or OpenEdge management at all**
- **Name Server no longer existing**

Classic AppServer Architecture



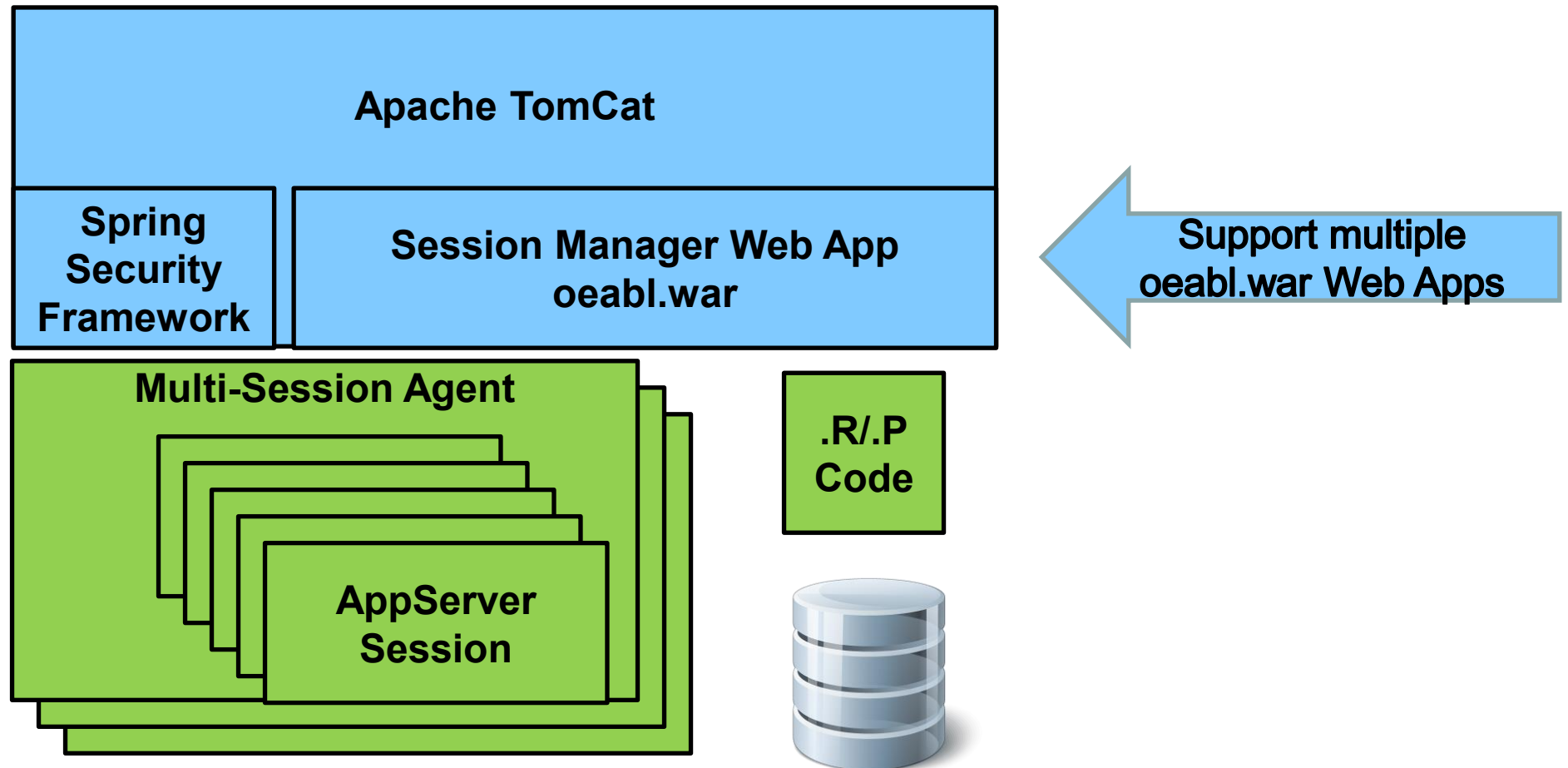
Classic AppServer Architecture



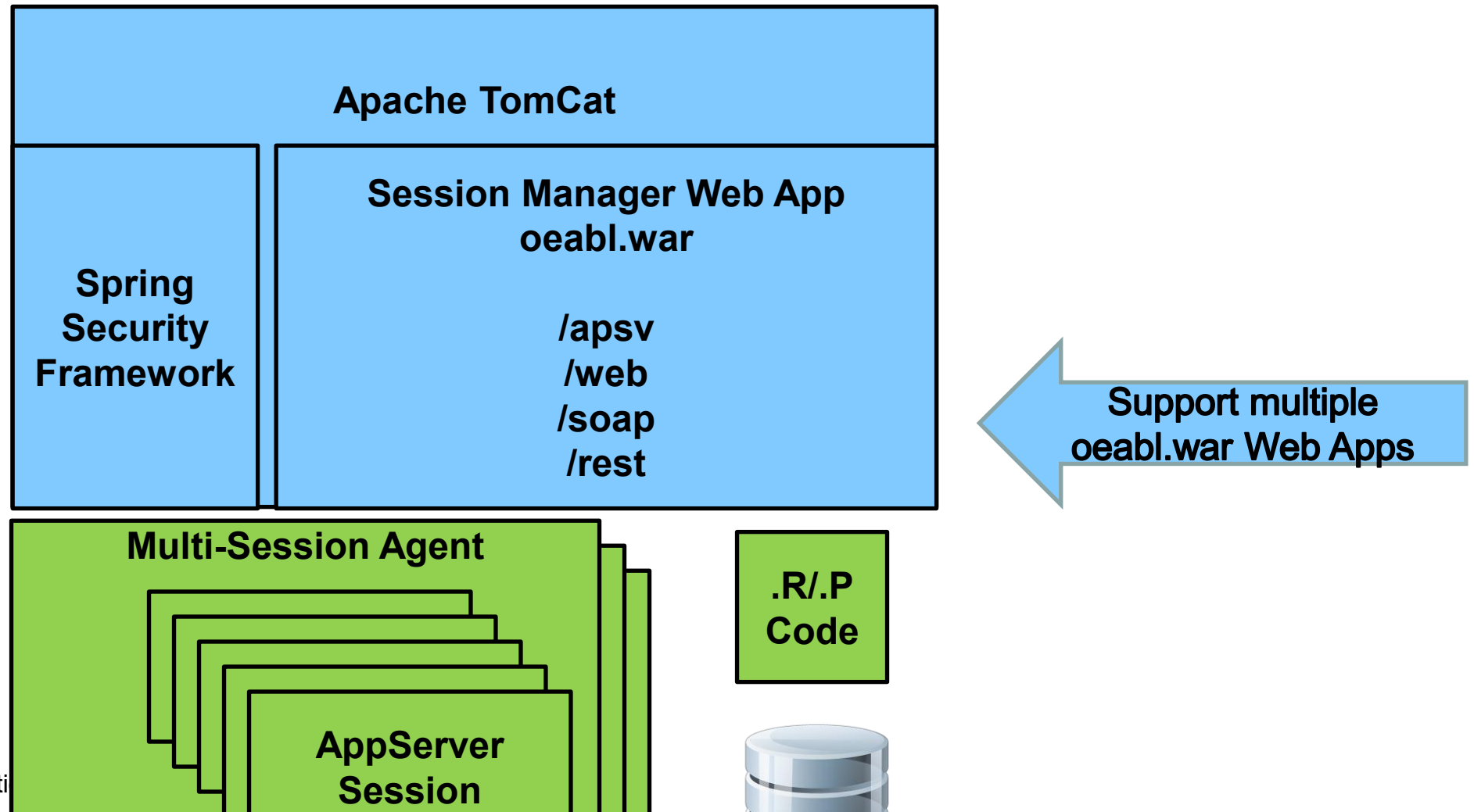
PASOE in contrast

- AppServer Broker replaced with Session Manager Tomcat webapp
- AppServer agent (C++) now supports multi-session/multiple threads
- Agent capable of executing multiple requests at a time – in parallel threads
- Agent capable of maintaining multiple sessions at a time (default configuration is 200 sessions per agent)
- To scale up (execute more concurrent requests) choice of using more agent processes or threads within the agents' processes
- Scaling more lightweight

PASOE Architecture



PASOE Architecture



„Session“

- ABL Session
 - The memory space allocated for an ABL runtime environment
 - All the programs (R-Code and Data allocated by that runtime)
 - -Bt and other memory related startup parameters per Session
 - Isolated from each other
 - Initial Sessions numbered 4, 7, 8, 9, 10, ...
 - ABL Sessions are executed by Threads within MS-Agent
 - PASOE supports multiple MS-Agents
 - PASOE can be configured to use 1 Session per MS-Agent only, or 100's
- Confusing term with Login Session, Session Management, etc.

PASOE – 3 license options

- **Production PASOE**
- **Development PASOE**
- **PASOE lite**
- Differences in scalability
- Production PASOE secure out of the box (all endpoints locked down)
- Development PASOE open out of the box (developer friendly)
- PASOE lite for “occasional users”, limited to 5 ABL sessions, one ABL application per PASOE instance

Java Version Dependency

- OpenEdge Installer requires JDK installed before OpenEdge install
- OpenEdge Installer asks for “*minimum*” Java version
- OpenEdge 12.2: Java 11
- OpenEdge 12.8: Java 17
- Recommended to stick with the latest patch level in that Java version
- OpenEdge 12.2 works with Java 13, but crashes with Java 17 ...

Home vs. Base (Instance)

- CATALINA_HOME: %dlc%\servers\pasoe
- CATALINA_HOME contains binaries – and should not be used as an AppServer
- CATALINA_BASE: an instance – contains specific configuration for an application, may contain application R-Code
- Instance can be copied to another PC by copying the instance folder
- %DLC%\servers\pasoe\bin\tcman to „create“ a new instance
- %CATALINA_BASE%\bin\tcman to manage the instance
 - start / stop / configure
 - deploy web applications

PASMAN vs. TCMAN

- TCMAN is the Tomcat management script, to create, start, stop, ... an PASOE instance
- TCMAN is **used from %DLC%\servers\pasoe\bin** to create PASOE instance
- TCMAN is **used from within the PASOE instance** to start, stop, manage an PASOE instance (e.g. deploy web applications)
- https://www.progress.com/docs/default-source/default-document-library/pasoe_tcman_reference.pdf

PASMAN vs. TCMAN

- PASMAN is a wrapper to TCMAN
- Within *proenv* in System PATH
- Requires `-l` parameter for instance options
- <https://docs.progress.com/bundle/pas-for-openedge-management-122/page/PASMAN.html>

Instance list

- %DLC%\servers\pasoe\conf\instances.windows
- \$DLC/servers/pasoe/conf/instances.unix
- Contains instance folders and instance logical names
- Duplicates not allowed
- Remove entries in case you have deleted a PASOE instance folder

Classic AppServer recap

- AppServer Broker (Java) manages pool of AppServer agents
- AppServer agent (C++) more or less like a Progress TTY client
 - Early versions of the AppServer agent could be launched directly; resulting in a TTY like session with a broken PROTERMCAP
- Agent capable of executing a single request at a time
- Agent contained a single session at a time
- To scale up (execute more concurrent requests) new Agent processes required to be started

Demo

- Create PASOE AppServer using TCMAN
- Start/Stop PASOE using TCMAN
- Register/Unregister as a Windows Service
- Review configuration files
- Connect from ABL client
- PASOE logfiles

Review openedge.properties

- [AppServer.SessMgr]
- [AppServer.Agent]
- [AppServer.Agent.<abl app name>]
- [AppServer.SessMgr.<abl app name>]
- [<abl app name>.<web app name>.<transport>] sections
- ABL Configuration
- PROPATH
- Startup Parameters
- Agent-Procedures

Review openedge.properties

- First ABL app is named like the PASOE instance
- Additional ABL apps are named differently
- `conf/openedge.properties`.README with detailed description of settings

PASOE Transports

- /apsv
 - Classic AppServer Protocol for OpenEdge and OpenClient based clients
 - Basically AIA http tunnel built into AppServer
 - No support for proprietary TCP/IP and UDP based connections
- /soap – Web Service Adapter of classic AppServer built in
- /rest – REST Adapter of classic AppServer built in
 - GET & POST, JSON payload
- /web
 - Replacement for WebSpeed
 - Foundation for more flexible (REST and SOAP) Web Services
 - **The best thing since sliced bread**

APSV Transport

- Transport for ABL Clients, Java or .NET OpenClients
- -URL `http://<server>:<port>/apsv`
- -URL `http://<server>:<port>/<abl application>/apsv`

Requirements for modern web apps

- Full page requests are the exception today
- JavaScript typically executed in the browser, single-page-applications
- Web controls like Kendo UI can perform Data Binding within the browser
- Data and page layout elements are retrieved in separate web requests
- Data retrieved through backend service calls
- Web frontend may perform multiple backend requests in parallel

WEB Transport

- Replacement for classic WebSpeed
- Designed to use ABL to build API's

Compatibility Web Handler

- Optional feature – depending on Web transport configuration
- Supports classic WebSpeed embedded SpeedScript and CGI Wrapper
- Typically, without or only little modification
- Support for mapped web objects limited, introduced in OpenEdge 12.8
 - Not available in OpenEdge 12.2
 - Execution of existing mapped web objects
 - Not able to rebuild .off files

Web Handler

- Web Handlers provide a very flexible way to handle web requests
- Synchronous request-response pattern
- Supports html page generation
- Supports service requests as well
- Flexible enough to provide an alternative to the REST Adapter and Web Services Adapter (SOAP)
- ABL classes, extending `OpenEdge.Web.WebHandler`

OpenEdge.Web.WebHandler

Method Summary

Options Name	
	INTEGER HandleDelete (IWebRequest)
	INTEGER HandleGet (IWebRequest)
	INTEGER HandleHead (IWebRequest)
A	INTEGER HandleNotAllowedMethod (IWebRequest)
A	INTEGER HandleNotImplemented (IWebRequest)
	INTEGER HandleOptions (IWebRequest)
	INTEGER HandlePatch (IWebRequest)
	INTEGER HandlePost (IWebRequest)
	INTEGER HandlePut (IWebRequest)
#	INTEGER HandleRequest ()
	INTEGER HandleTrace (IWebRequest)

Web Handler

- WebSpeed in PASOE provides request handler mapping out of the box (classic Web Speed required customization of web-disp.p for this)
- Based on configuration in openedge.properties
- New PDSOE project type ABL Web Application creates and registers a single handler
- Additional handlers can be set up in OpenEdge Management or configuration files

URL Mapping

- Configuration based
- Tomcat parses request URI for patterns
- `http://localhost/web/Customer/1`
- More „rest-style“ URI's
- Higher priority in search engines compared to `http://localhost/cgi-bin/cgiip.exe/Customer.w?CustNum=1`
- Web handler are specialized ABL classes

URL Mapping – in conf/openedge.properties

openedge.properties ✖

```
[oepas1.ROOT.WEB]
adapterEnabled=1
defaultCookieDomain=
defaultCookiePath=
defaultHandler=OpenEdge.Web.CompatibilityHandler
handler1=JsonDataHandler: /JsonData/{BusinessEntityName}
handler2=CustomerListWithSearchHandler: /CustomerSearch
handler3=CustomerHandler: /Customer/{CustNum}
handler4=DemoHandler: /Demo
handler5=nullHandler: /Ab1WebAppProject
handler6=nullHandler: /Data
srvrDebug=0
```

**Not gaps or
duplicates
allowed in
handler list!!!**

URL Mapping in .handlers file

- JSON file in /webapps/ROOT/WEB-INF/adapters/web/ROOT/ROOT.handlers

```
{  
  "version": "2.0",  
  "serviceName": "",  
  "handlers": [  
    {  
      "class": "Test.PASOE.SampleWebHandler",  
      "uri": "/Orders/{CustomerNum}",  
      "enabled": true  
    },  
    {  
      "class": "Test.PASOE.SampleWebHandler",  
      "uri": "/Orders",  
      "enabled": true  
    }  
  ]  
}
```

Sample web handler

```
CustomerHandler.cls X
39 METHOD OVERRIDE PROTECTED INTEGER HandleGet (poRequest AS OpenEdge.Web.IWebRequest):
40
41     DEFINE VARIABLE response AS OpenEdge.Web.WebResponse NO-UNDO.
42     DEFINE VARIABLE writer AS OpenEdge.Web.WebResponseWriter NO-UNDO.
43
44     DEFINE VARIABLE jsonObject AS JsonObject NO-UNDO.
45
46     DEFINE VARIABLE iCustNum AS INTEGER NO-UNDO.
47     DEFINE VARIABLE cCustNum AS CHARACTER NO-UNDO.
48
49     EMPTY TEMP-TABLE ttCustomer .
50
51     response = NEW WebResponse().
52     writer = NEW WebResponseWriter(response).
53
54     cCustNum = poRequest:GetPathParameter("CustNum") .
55
56     ASSIGN iCustNum = INTEGER (cCustNum) NO-ERROR .
```

```
jsonObject = NEW JsonObject().
jsonObject:READ(TEMP-TABLE ttCustomer:HANDLE).

response:StatusCode = 200.
response:ContentType = "application/json".

writer:Write(jsonObject:GetJsonText()).

writer:Flush().
writer:Close().

RETURN 0.
```

Using parameters

- Access to path parameters:

```
poRequest:GetPathParameter ("CustNum")
```

Demo Web handler

- Review sample web handler
- Register in .handlers file
- Execute requests

WebHandler tuning

- By default, OpenEdge logging framework initialized for every request, even when not used
- Create logging.config file anywhere in PROPATH
- Gains 8-15 msec per request (YMMV)

```
logging.config ×
smartpas_stream > openedge > logging.config
1 {
2   "logger": {
3     "OpenEdge.Web.InternalWebRouter": {
4       "logLevel": "OFF",
5       "filters": [
6         "VOID_WRITER"
7       ]
8     }
9   }
10 }
```

Additional PASOE Web Apps

- `oeabl.war` – support multiple ABL application in a single PASOE instance
- `oemanager.war` – management API for PASOE
- `oedbg.war` – debugger support for PASOE
- `oehealth.war` – health check web application
- `oediagstore.war` – storage for profiler logfiles in an OpenEdge DB

Demo additional Web Apps

- Deploy additional oeabl.war
 - Review openedge.properties
- Deploy oemanager.war
- Deploy oedbg.war

Demo oemanager

- Swagger / OpenAPI reference
- Execute requests in browser
- Execute requests from OpenEdge application

oemanager VS Code Extension

The screenshot shows the Visual Studio Code interface with the 'Extension: PASOE Manager Extension' panel open. The left sidebar shows the 'EXTENSIONS' view with a search bar and a list of installed extensions. The main panel displays the details for the 'PASOE Manager Extension' by Consultingwerk Application Modernization Solutions Ltd. The extension is installed and its details are shown, including a description, features, and installation information.

EXTENSIONS

Search Extensions in Marketplace

INSTALLED (44)

- Jira and Bitbucket (Atlassian Labs)** (257ms)
Bringing the power of Jira and Bitbucket to VS C...
Atlassian
- Light High Contrast Theme**
Based on Light+ but with borders and black text
Stephen Wassell
- OpenAPI (Swagger) Editor** (2094ms)
OpenAPI editing, validation and preview in VS C...
42Crunch
- OpenEdge ABL** (103ms)
OpenEdge ABL support for VS Code
Riverside Software
- OpenEdge ABL Formatter** (25ms)
Progress OpenEdge (ABL) code formatter
Baltic Amadeus
- PASOE Manager Extension**
VS Code Extension to maintain PASOE Instances...
Consultingwerk Application Modernization Sol...
- Perforce for VS Code** (48ms)
(2020 Fork) Perforce integration with VS Code's ...
mjcrouch

Extension: PASOE Manager Extension

PASOE Manager Extension
Consultingwerk Application Modernization Solutions Ltd. | 149 | ☆☆☆☆☆
VS Code Extension to maintain PASOE Instances using the OE Manager Rest Interfaces (<https://docs.progress.com/de-DE...>)

Disable **Uninstall** ☐ Auto Update

DETAILS **FEATURES** **CHANGELOG**

PASOE Manager Extension

Description
A VS Code extension for Progress Application Server for OpenEdge (PASOE).
The extension contains the following register commands:

- PASOE: List of Pasoe Agents**
Shows the available PASOE agents in the OUTPUT console.
- PASOE: List of Agent Sessions**
Displays a list of the available agent sessions in the OUTPUT console.
- PASOE: Trim Agents**

Installation

Identifier	consultingwerkapplicati
Version	1.2.0
Last Updated	2025-03-19, 17:05:47
Size	294.41MB

Marketplace

Published	2025-01-09, 16:11:49
Last Released	2025-03-19, 15:22:23

Categories

OUTPUT GITLENS SPELL CHECKER PROBLEMS (602) DEBUG CONSOLE TEST RESULTS TERMINAL ...

Filter (e.g. text, **/*.ts, !**/node_modules/**) Showing 0 of 602

No problems have been detected in the current file.

oemanager VS Code Extension

- <https://marketplace.visualstudio.com/items?itemName=Consultingwerk.ApplicationModernizationSolutionsLtd.oemanager>
- Typical PASOE interactions during development
- Agent status
- Agent sessions
- Trim Agents
- PING Agents

SOAP and REST Transports

- Web Services Adapter (OpenEdge 10) and REST Adapter (OpenEdge 11.2) integrated into PASOE
- No more deployed in separate Tomcat (or Java AppServer or your choice)
- May require HTTP proxy in DMZ as PASOE typically inside the network
- Web Service Adapter requires ProxyGen to build WSM file
- REST Adapter requires PDSOE to build paar file
- Changes in URL require updates of clients, or use of URL rewrites in PASOE

SSL / TLS

- `conf/catalina.properties`
- `conf/server.xml` -> `SSLHostConfig`

Authentication options

- None = anonymous
- Local = Tomcat user's file
- LDAP and Active Directory
- OE-Realm = ABL Class for Authentication
- STS = PASOE as Authentication Gateway

Authentication options

- Basic authentication
- Form based authentication
- SSO
- OAuth2
- SAML

PASOE challenges – strategy for request execution

- Classic AppServer: round-robin, attempt to spread load over all available agents
- PASOE: first available session (in order they were started) gets used the most
- Idea is to reduce memory consumption as fewer sessions are needed when there is not a lot of load, reduce initialization time of sessions (session startup procedure)
- Challenge is that memory leaking ABL code has more severe impact in sessions

PASOE challenges

- Agent process memory consumption is the sum of all session's memory
- Agent crash due to memory issues impacts all sessions in the agent process

Session Manager Configuration

```
numInitialAgents=1
```

Number of multi-session agents to start when AppServer is started.

```
maxAgents=10
```

Maximum number of concurrent multi-session agents that can run on the AppServer.

```
maxABLSessionsPerAgent=200
```

Maximum number of concurrent sessions a multi-session agent can run concurrently.

```
maxConnectionsPerAgent=16
```

Maximum number of tcp connections between session manager and multi-session agent.

```
numInitialSessions=5
```

Indicates the number of initial session that are pre- started by the Application Server.

Session Manager Configuration

- Higher costs for OS calls
- OS-COMMAND and INPUT THROUGH seem to require delays for thread-synchronization and forking of the OS environment
- Potentially also high requirement for OS memory on Linux when forking shell process from MS-Agent (INPUT-THROUGH, OS-COMMAND, ...)

SmartUpdate

Stay Tuned for the Latest News,
Product Updates, and Event Highlights!

Subscribe Now to Consultingwerk's Official Newsletter!

www.consultingwerk.com/newsletter
or marketing@consultingwerk.com

- October 9th - Implementing MCP Servers With OpenEdge
- Integrate Chatbots Into OpenEdge Applications
- Using Windsurf IDE As An AI Coding Assistant For OpenEdge

UPCOMING
WEBINARS

Questions



Consultingwerk

software architecture and development