

ABL Memory Profiler: Are You Leaking Memory?

Sunil Jardosh

Progress Software

September 2024



All roadmaps are for informational purposes only, and the reader is hereby cautioned that actual product development may vary significantly from roadmaps.

These roadmaps may not be interpreted as any commitment on behalf of Progress, and future development, timing and release of any features or functionality described in the roadmaps remains at our sole discretion.



Agenda

- Understanding ABL Memory Use
- Memory Profiler Overview
- Demo
- Architecture and Configuration
- Memory Profiler Product Vision
- Next Steps
- Questions

Understanding the Problem

- “Resolving performance-related problems in ABL applications is very difficult, especially when it comes to memory consumption.”
- **How do we know that?**
 - Analysis of technical support tickets related to resource consumption
 - Community interactions
 - We have performance profiling capability, but the CPU is only part of the problem

To Address the Problem

- Historically, OpenEdge added tools for Memory Use analysis of OpenEdge Apps
 - Dynamic Object Logging
 - PAS for OpenEdge, OE JMX
 - Pulse Monitoring Tool

Memory Use in an ABL App

Memory Use In An ABL App

Platform Memory

- Examples:
 - -Bt – Temp-table Buffer Pool
 - -s – Runtime Stack
 - -mmax – R-Code Memory
 - -D – Directory Entries
 - -reusableObjects – OO Cache
- Some Control by Startup Parameters

Application Memory

- .p Procedures
- OOABL Objects & .NET
- Built-in OO Objects
- Database Connections
- Dynamic Temp-tables, Buffers, Datasets, Queries, etc.
- Dynamic UI Widgets
- Memptrs, Longchars
- Servers, Sockets, Async Objects,...

The Unanswered Questions About Memory

- **Where is my Application Leaking Memory?**
- **Where is all the Memory going? In other words,...**
 - What got allocated?
 - Where was it allocated?
 - When was it allocated?
 - How big is it?

For all of this, you need Data... Lots of Data

What Data Is Needed?

For *Every Single Application Object* your App Creates...

- .p Procedures
- OOABL Objects & .NET
- Database Connections
- Dynamic Temp-tables, Buffers, Datasets, Queries, etc.

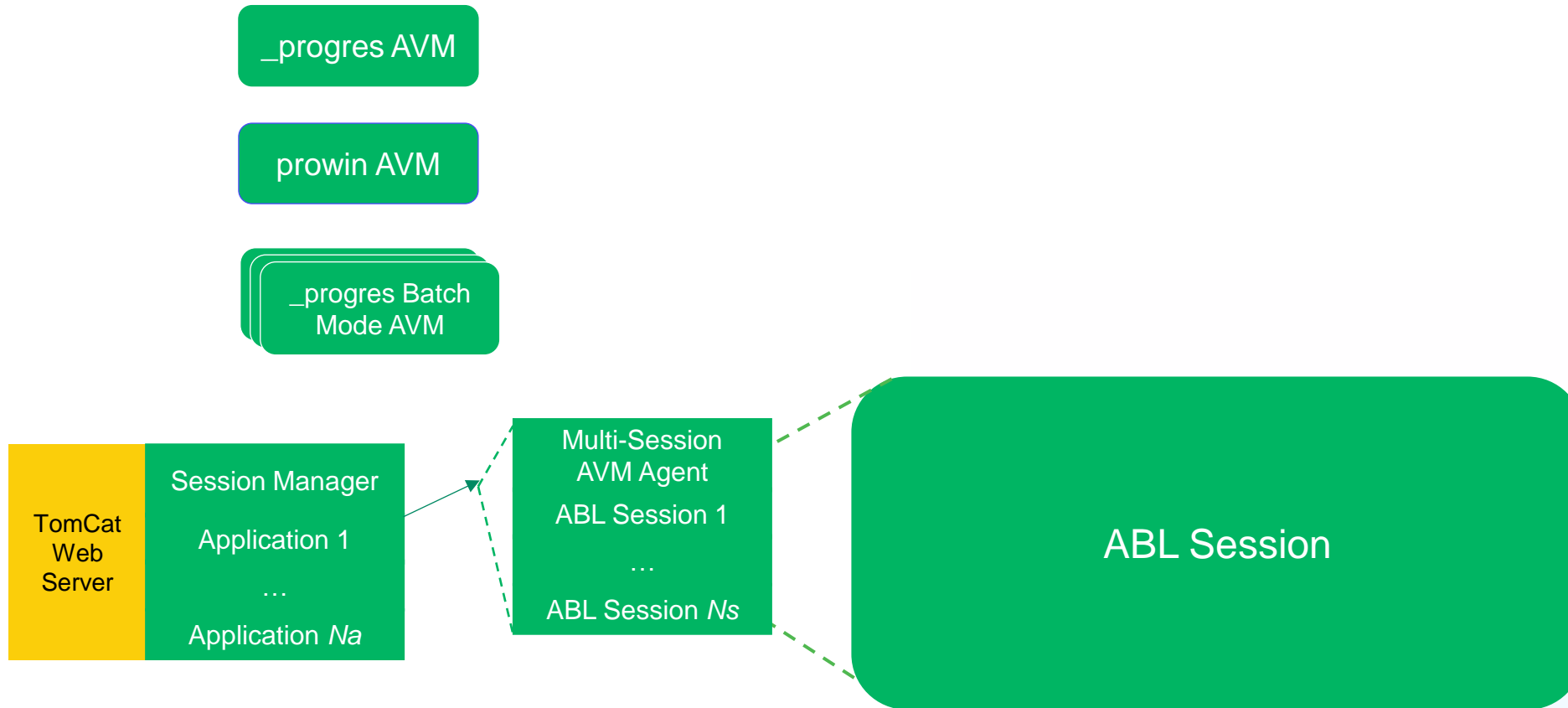
The AVM needs to collect what Objects are Alive at any time, and...

- What each Object is (its name, type)
- How much memory it's using
- When it was created
- Where it was created (Call-Stack)
- Where the Object is Scoped (if it is scoped)

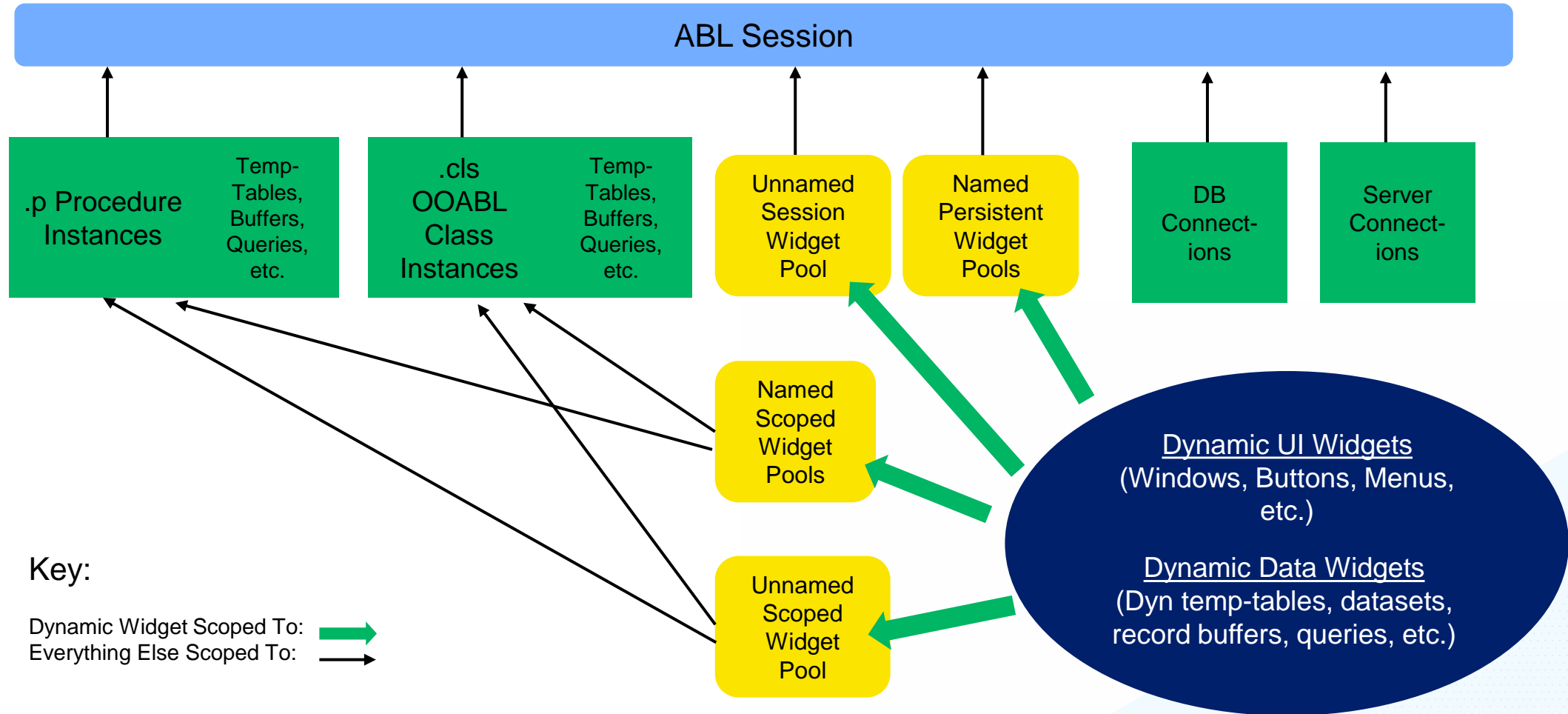
Memory Profiling Overview

ABL Memory Profiling Overview

OpenEdge AVM Processes

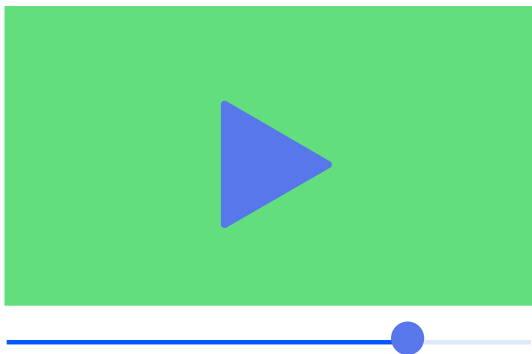


Types of Application Objects and Their Scope

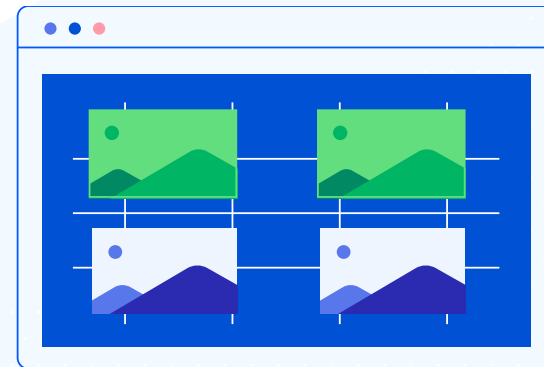


Memory Use In Your App Changes Over Time...Your Memory Profile Is Time Series Data

A Stream of Data Telling the Story of the App's Allocations and Deallocations



A Series of Snapshots at Regular Time Intervals



Memory Profiling Configuration

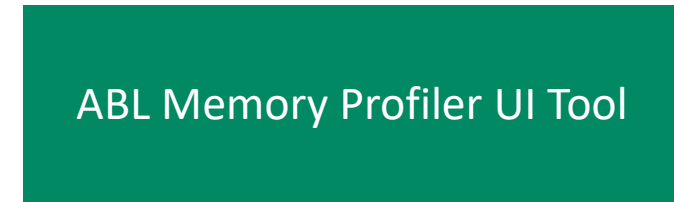
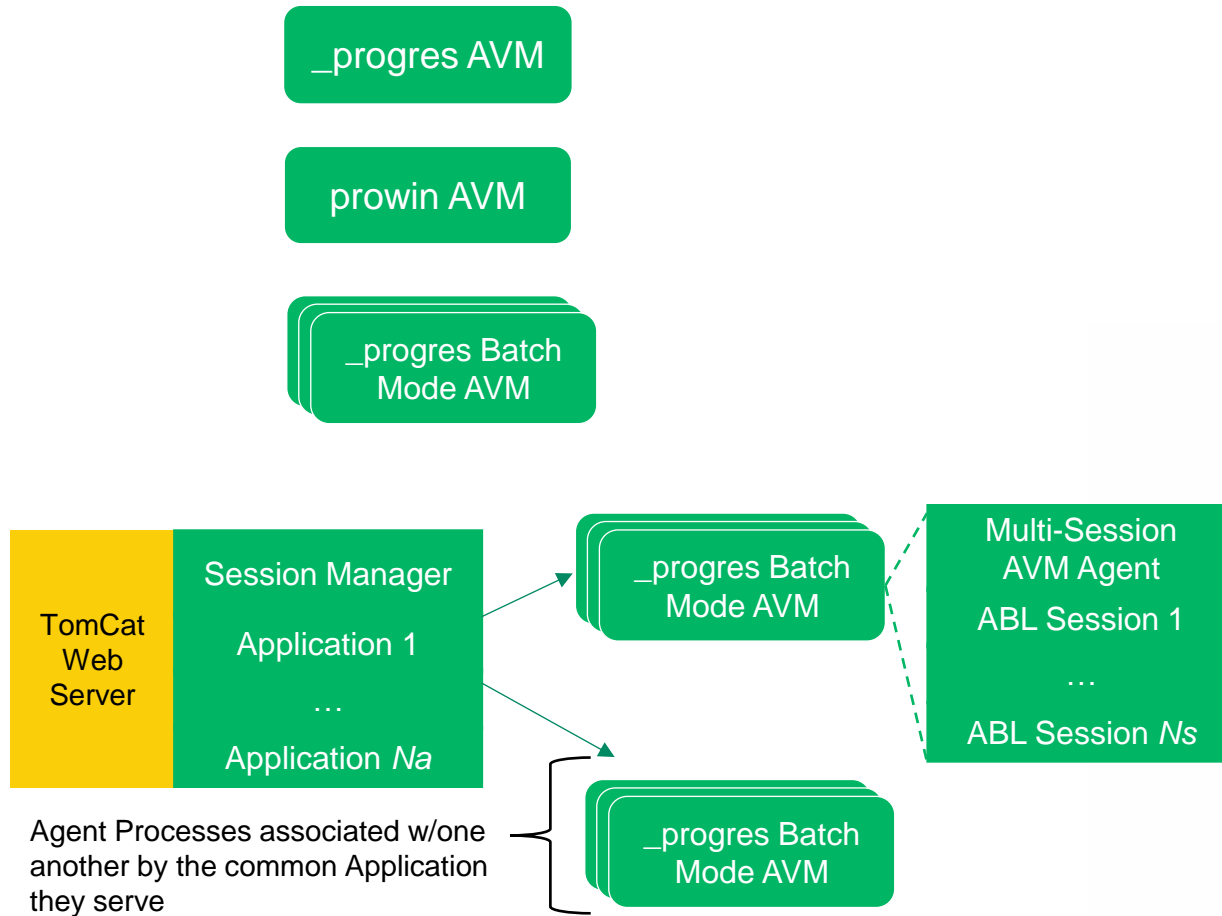
- Startup parameter
- Configuration file
- Language extension
- Snapshot File

Demo

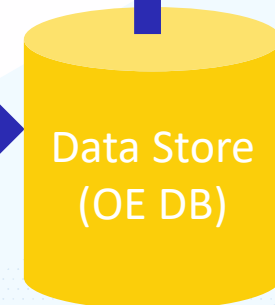
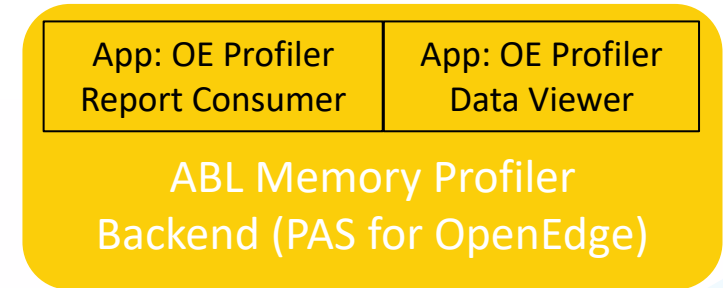
Architecture

ABL Memory Profiling Architecture

OpenEdge AVM Processes

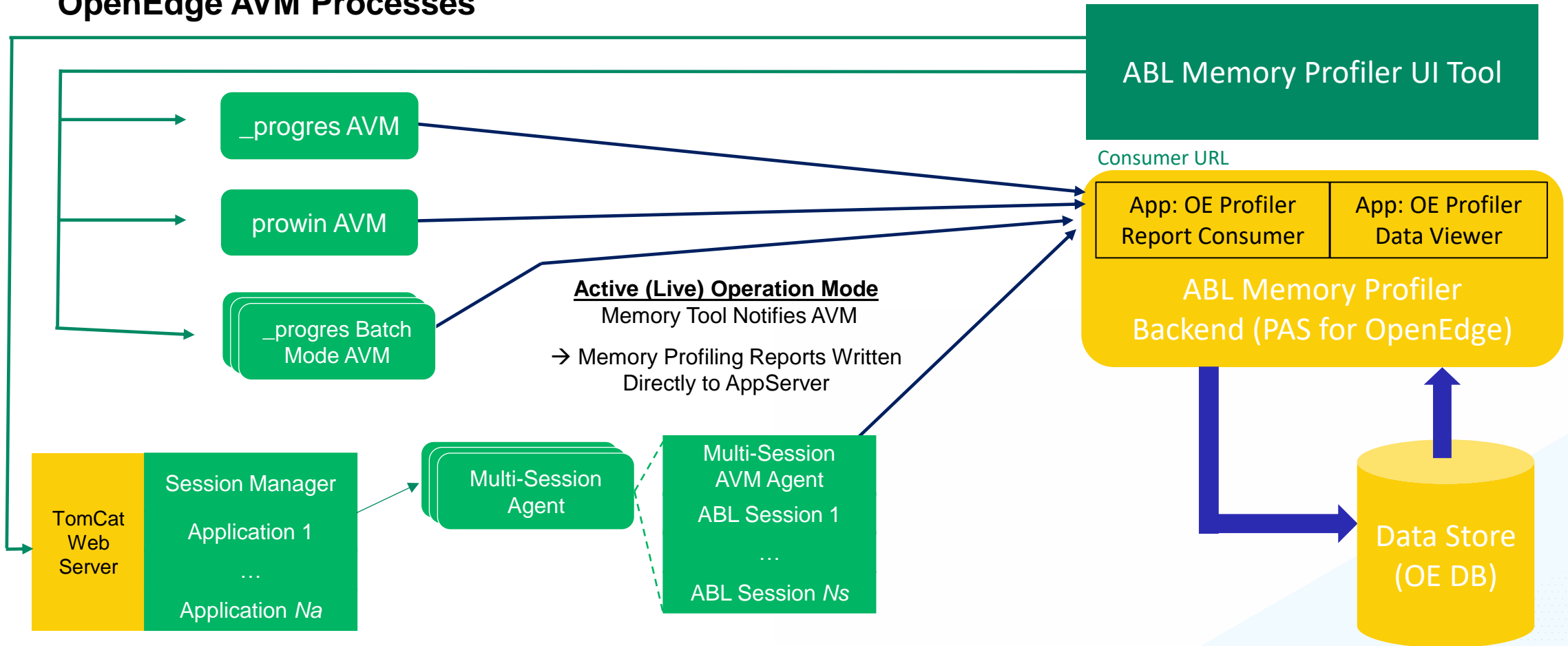


Consumer URL



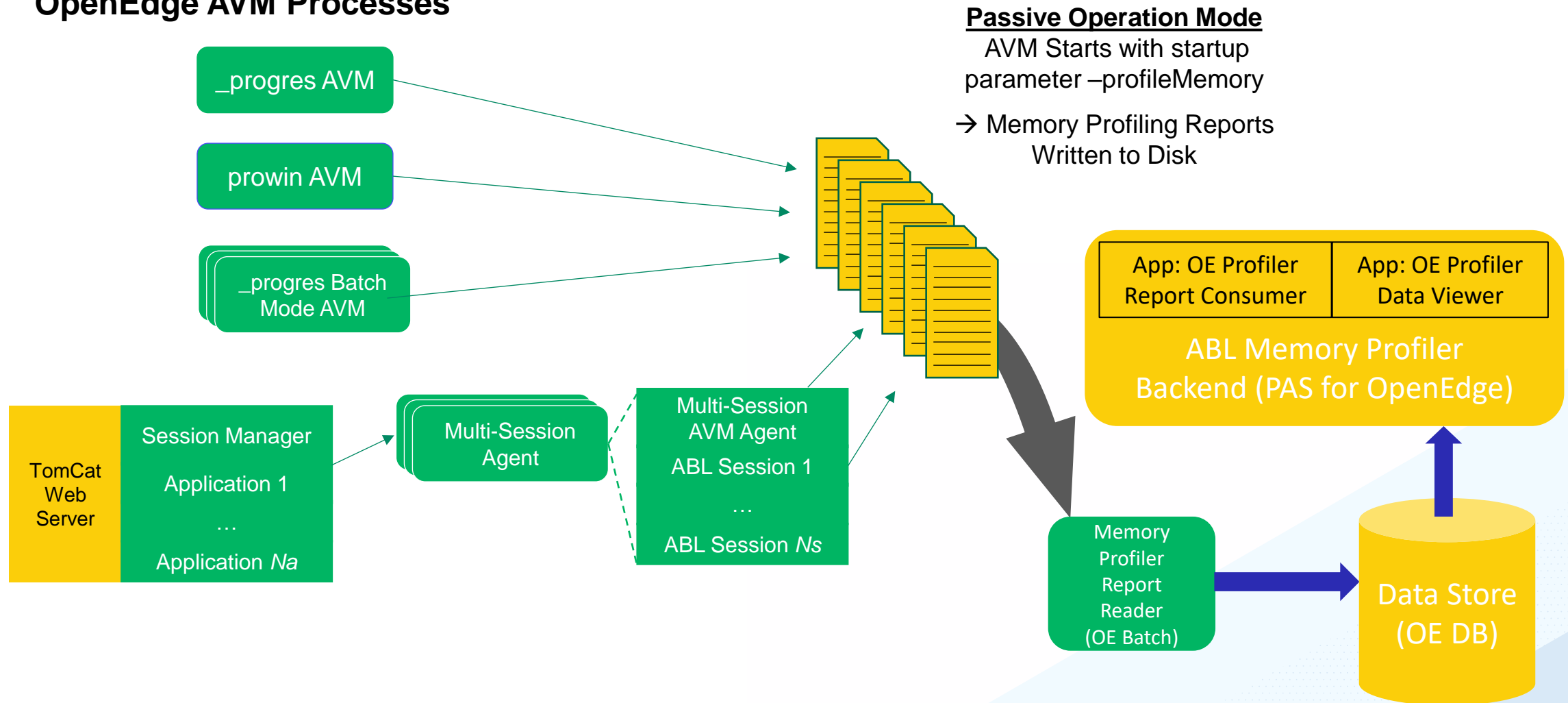
ABL Memory Profiling Architecture

OpenEdge AVM Processes



ABL Memory Profiling Architecture

OpenEdge AVM Processes



Passive Operation Mode

AVM Starts with startup parameter `-profileMemory`
→ Memory Profiling Reports Written to Disk

Memory Profiler Tool Product Vision

Profiler Product Vision

Purpose

- Identify memory leaks
- Benchmarking
- Review application architecture

Persona

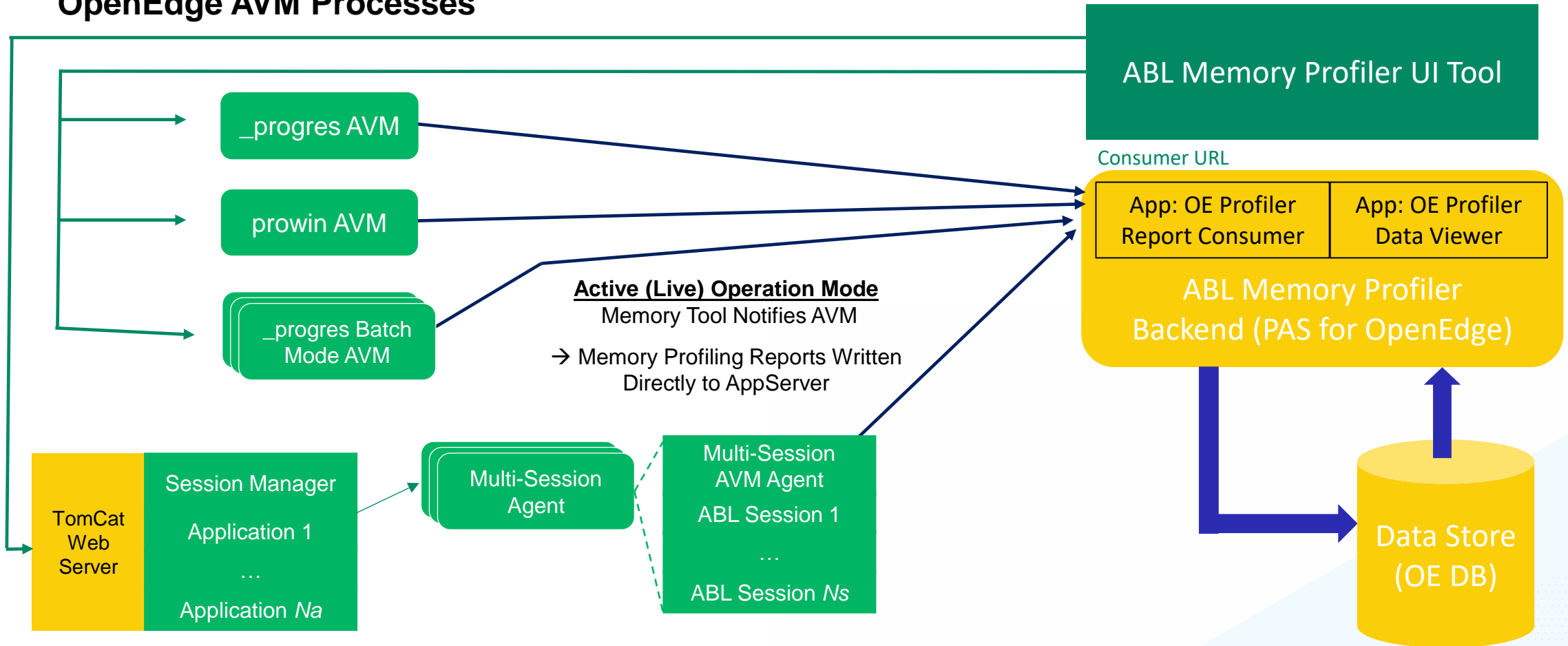
- Developer

Scope

- Development
- Production (recording only)

ABL Memory Profiling Architecture

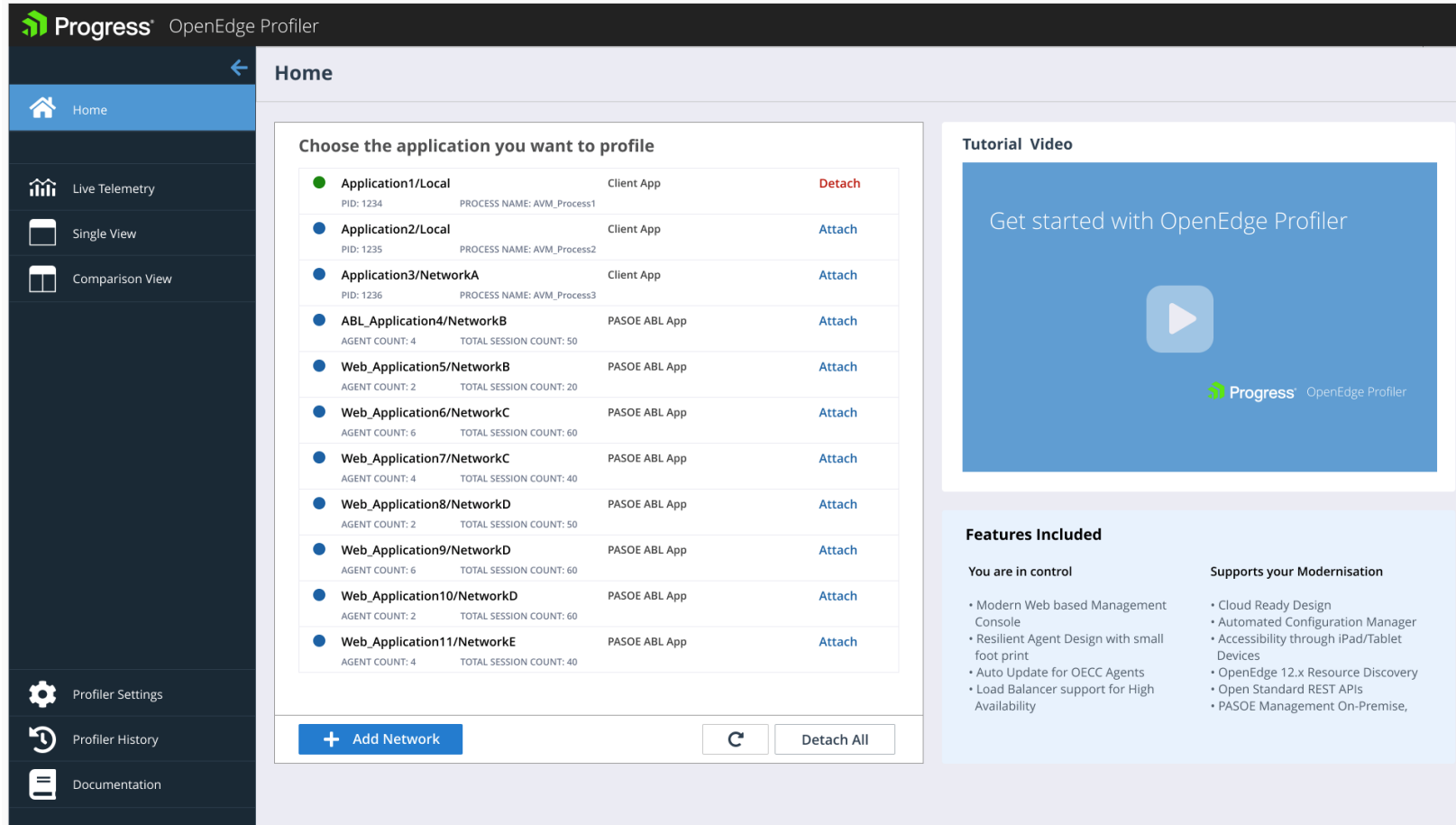
OpenEdge AVM Processes



Profiler Product Vision

Ways to Profile

- Attach to a PAS for OpenEdge ABL App or a standalone AVM for live profiling
- Start/Stop profiling through CLI or ABL code
- Import a profiling session stored to disk



The screenshot displays the Progress OpenEdge Profiler web interface. The top navigation bar includes the Progress logo and the text "OpenEdge Profiler". Below this is a "Home" header with a back arrow. A left sidebar contains navigation options: Home, Live Telemetry, Single View, Comparison View, Profiler Settings, Profiler History, and Documentation. The main content area is titled "Choose the application you want to profile" and contains a table of applications. At the bottom of the table are buttons for "+ Add Network", "C", and "Detach All". To the right of the table is a "Tutorial Video" section with a play button and the text "Get started with OpenEdge Profiler". Below the video is a "Features Included" section with two columns of bullet points.

Application	Type	Action
Application1/Local PID: 1234 PROCESS NAME: AVM_Process1	Client App	Detach
Application2/Local PID: 1235 PROCESS NAME: AVM_Process2	Client App	Attach
Application3/NetworkA PID: 1236 PROCESS NAME: AVM_Process3	Client App	Attach
ABL_Application4/NetworkB AGENT COUNT: 4 TOTAL SESSION COUNT: 50	PASOE ABL App	Attach
Web_Application5/NetworkB AGENT COUNT: 2 TOTAL SESSION COUNT: 20	PASOE ABL App	Attach
Web_Application6/NetworkC AGENT COUNT: 6 TOTAL SESSION COUNT: 60	PASOE ABL App	Attach
Web_Application7/NetworkC AGENT COUNT: 4 TOTAL SESSION COUNT: 40	PASOE ABL App	Attach
Web_Application8/NetworkD AGENT COUNT: 2 TOTAL SESSION COUNT: 50	PASOE ABL App	Attach
Web_Application9/NetworkD AGENT COUNT: 6 TOTAL SESSION COUNT: 60	PASOE ABL App	Attach
Web_Application10/NetworkD AGENT COUNT: 2 TOTAL SESSION COUNT: 60	PASOE ABL App	Attach
Web_Application11/NetworkE AGENT COUNT: 4 TOTAL SESSION COUNT: 40	PASOE ABL App	Attach

Tutorial Video
Get started with OpenEdge Profiler

Features Included

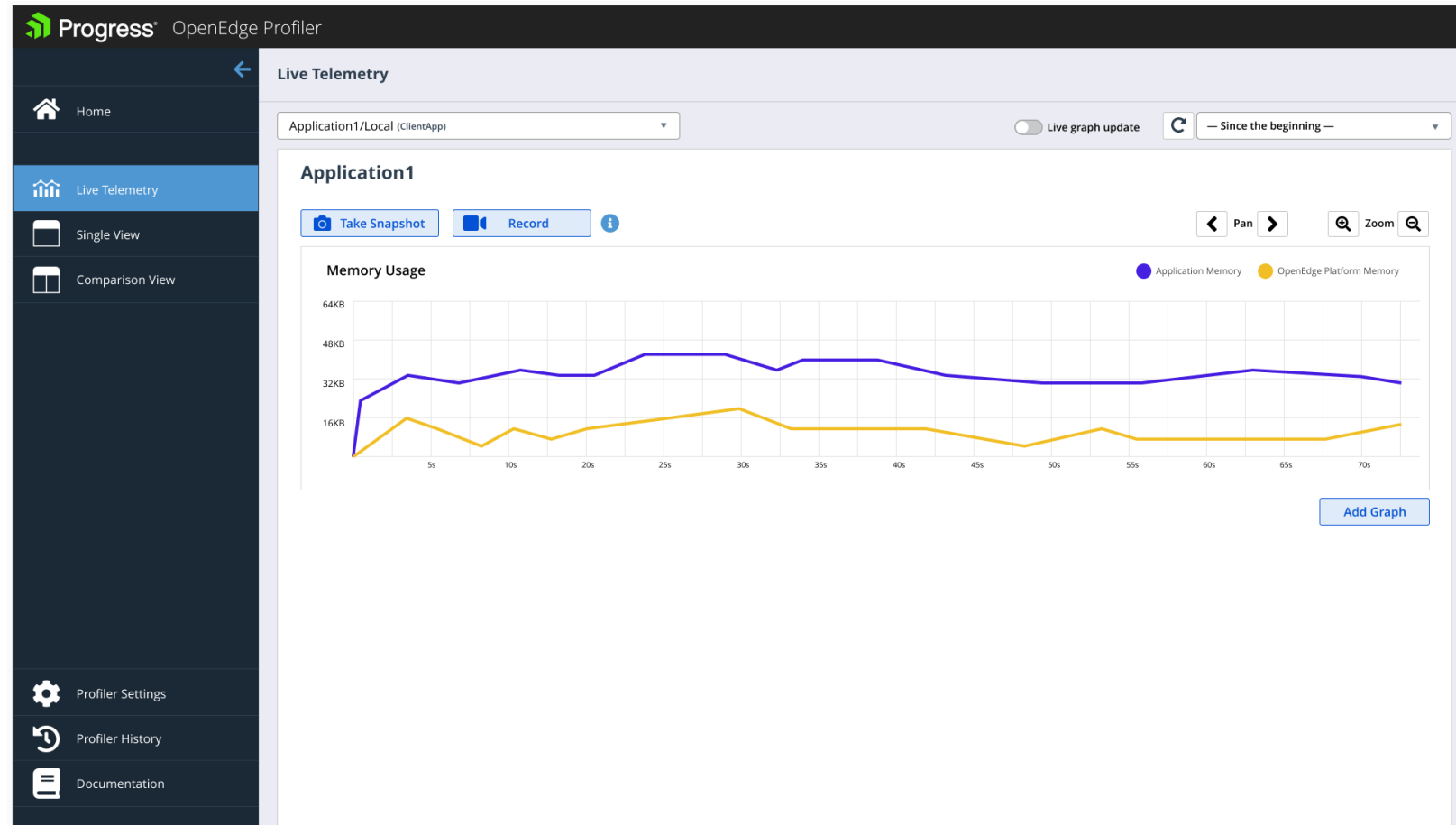
You are in control <ul style="list-style-type: none">• Modern Web based Management Console• Resilient Agent Design with small foot print• Auto Update for OECC Agents• Load Balancer support for High Availability	Supports your Modernisation <ul style="list-style-type: none">• Cloud Ready Design• Automated Configuration Manager• Accessibility through iPad/Tablet Devices• OpenEdge 12.x Resource Discovery• Open Standard REST APIs• PASOE Management On-Premise,
--	---

Profiler Product Vision

Connect Only (Live)

Once connected to an application, you can see some basic graphs showing the live memory allocations

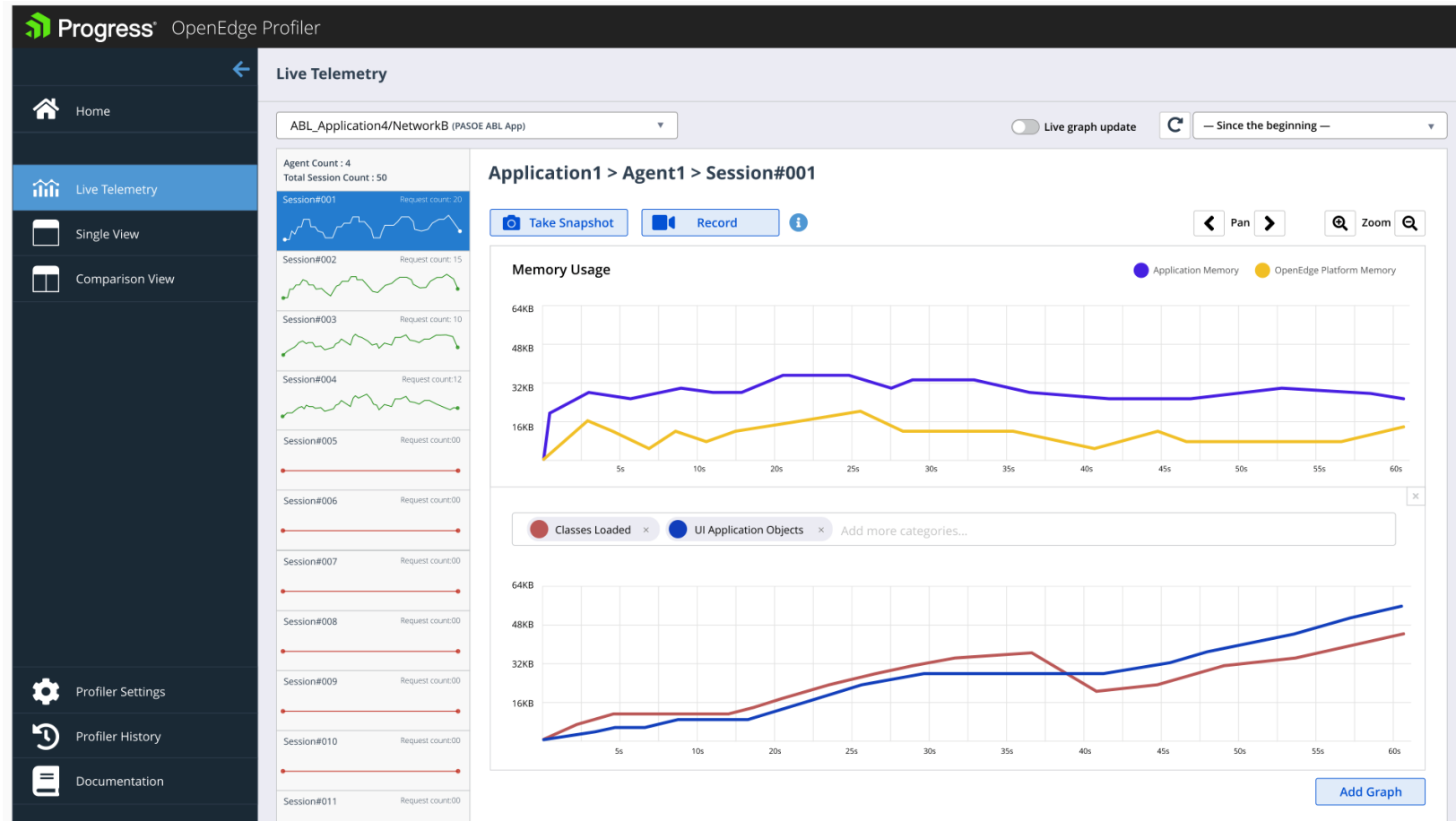
- Here you can see the memory occupied by a standalone ABL client over time, divided into application and platform memory



Profiler Product Vision

Connect Only (Live)

- Here you can see the memory occupied over time by a session of a PAS for OpenEdge ABL app
- Another graph added to the dashboard has only some filtered elements of ABL application

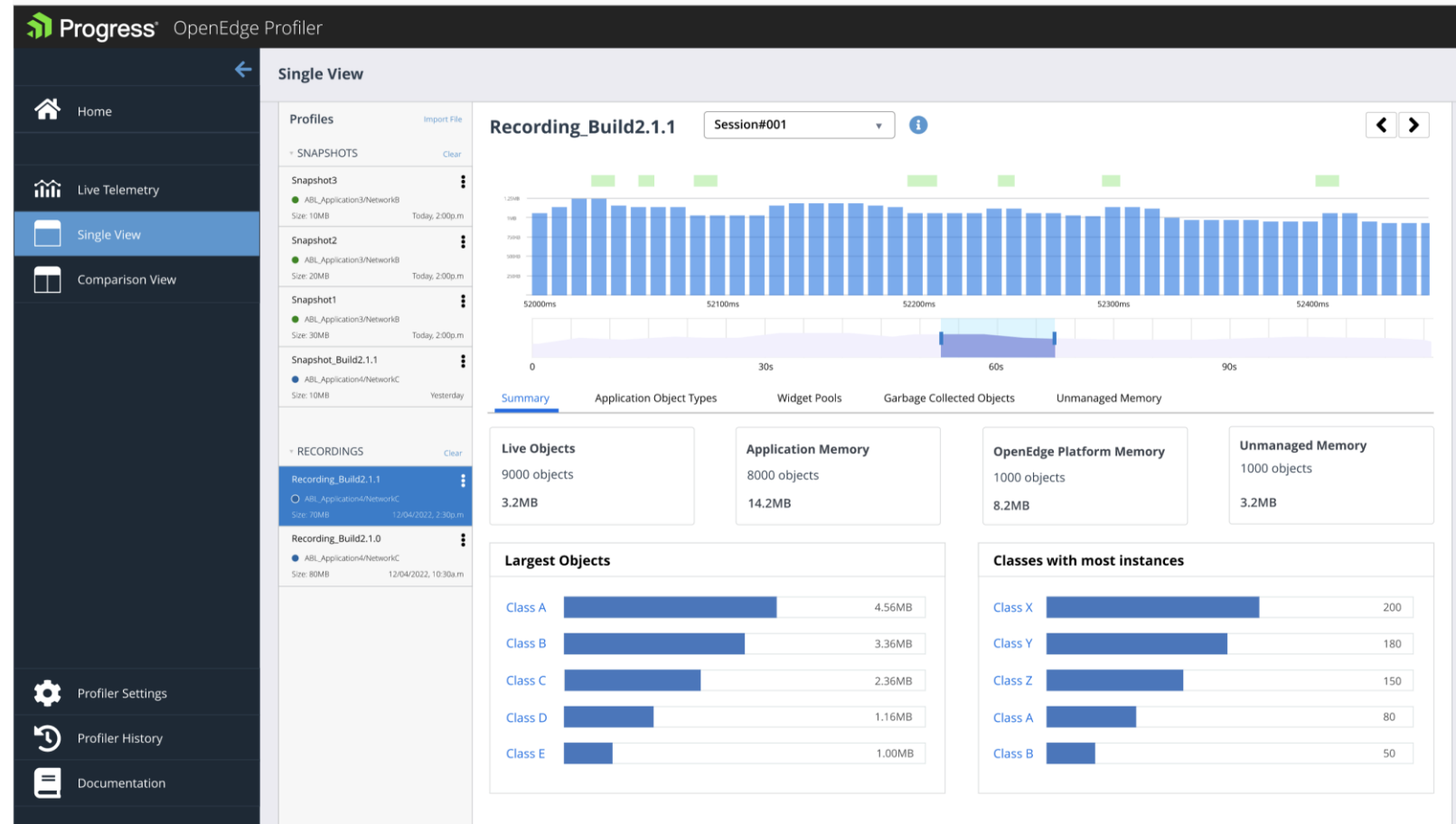


Profiler Product Vision

Summary View

Recordings will have a summary view to provide an overview

- Here you can see the summary of memory allocation for a window of time frame selected in a recording

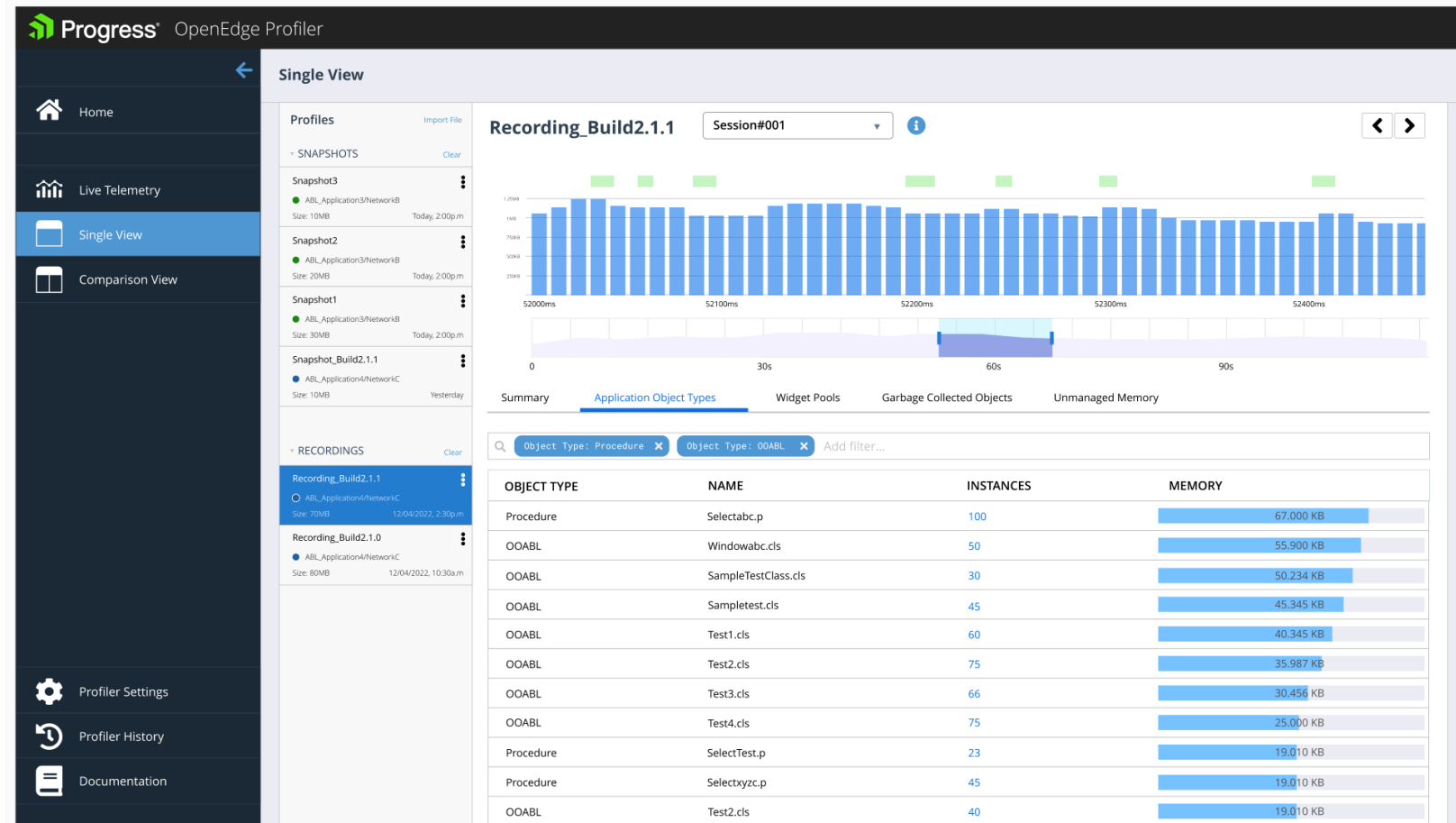


Profiler Product Vision

Recordings

View of the Application Object Types at this time in the recording

- Here you can see a window of time interval selected for a recording. The view below shows the memory allocation for the filtered ABL types within that time interval.



Profiler Product Vision

Snapshots

Snapshots are like a picture that captures the memory allocation at a moment in time

- Here you can see some filtered list of ABL types sorted by memory in a snapshot

The screenshot displays the Progress OpenEdge Profiler interface. The left sidebar contains navigation options: Home, Live Telemetry, Single View (selected), Comparison View, Profiler Settings, Profiler History, and Documentation. The main area is titled 'Single View' and shows a list of profiles and recordings. The 'Snapshots' section is expanded, showing 'Snapshot3' selected. The 'Application Object Types' tab is active, displaying a table of object types filtered by 'Procedure' and 'OOABL'. The table columns are OBJECT TYPE, NAME, INSTANCES, and MEMORY. The memory values are represented by blue bars.

OBJECT TYPE	NAME	INSTANCES	MEMORY
Procedure	Selectabc.p	100	67,000 KB
OOABL	Windowabc.cls	50	55,900 KB
OOABL	SampleTestClass.cls	30	50,234 KB
OOABL	SampleTest.cls	45	45,345 KB
OOABL	Test1.cls	60	40,345 KB
OOABL	Test2.cls	75	35,987 KB
OOABL	Test3.cls	66	30,456 KB
OOABL	Test4.cls	75	25,000 KB
Procedure	SelectTest.p	23	19,010 KB
Procedure	Selectbyzc.p	45	19,010 KB
OOABL	Test2.cls	40	19,010 KB
OOABL	Test3.cls	40	19,010 KB
Procedure	Testbyz.p	40	19,010 KB
Procedure	Testefg.p	40	19,010 KB

Profiler Product Vision

Drill Down Further

For any particular ABL type you can drill down to get more information on it

- Here you can see an interactive view that opens when clicking on the instances to show details of the instances. For every instance you can view its Call-Stack

The screenshot displays the Progress OpenEdge Profiler interface. On the left, a sidebar contains navigation options: Home, Live Telemetry, Single View (selected), Comparison View, Profiler Settings, Profiler History, and Documentation. The main area is divided into 'Profiles' and 'Recordings' sections. The 'Recordings' section is expanded to show 'Recording_Build2.1.1' selected. On the right, a window titled 'Instances of Windowabc.cls' is open, displaying a table of instance details and a call stack.

OBJECT ID	Memory Size	Age	Where Created	Line number	
234567	11.00KB	21:54:10.34	OOSamples.DataManager.DataReader()	ClassABC.cls	156
345679	9.78KB	21:45:05.56	OOSamples1.DataManager.DataReader()	Class098.cls	100
345674	9.23KB	21:25:09.67	OOSamples.EmployeeManager.Employee()	Class876.cls	80
456789	8.97KB	21:10:04.89	OOSamples.EmployeeManager.EmployeeClien...	Class234.cls	120
098876	8.01KB	20:58:45.00	OOSamples.EM.EmployeeClient()	ClassABC.cls	45
876532	7.80KB	22:09:45.50	OOSamples.EM.EmployeeClient()	Class098.cls	78
234567	7.65KB	20:45:56.06	OOSamples.EM.EmployeeClient()	ClassEFG.cls	90
245689	6.42KB	20:00:00.04	OOSamples.EM.EmployeeClient()	ClassABC.cls	86

NAME	LINE NUMBER	SOURCE FILE
Application.OOABLCClass.Execute.Run()	156	\\Source\OOSample\EmployeeList.cls
Application.OOABLCClass.Instance.Run()	80	\\Source\OOSample\Employee.p
Application.OOABLCClass.Render.Instance()	80	\\Source\OOSamples123\Employeees.p
Application.OOABLCClass.Render1.Instance()	450	\\Source\OOSample\EmployeeList.cls
Application.OOABLCClass.Instance.Run()	450	\\Source\OOSample\EmployeeList.cls
Application.OOABLCClass.Instance.Run()	450	\\Source\OOSample\EmployeeList.cls

Profiler Product Vision

Compare Snapshots

Compare snapshots taken at different times

- From a list of available snapshots, you can select any two for comparison and inspect the differences

The screenshot displays the Progress OpenEdge Profiler interface in 'Comparison View'. On the left, a sidebar lists 'Snapshots' and 'Recordings'. The main area shows 'Snapshot3' selected, with a warning message: 'Select another profile from the left to compare'. Below this, a table lists application object types with their names, instance counts for baseline and difference, and memory usage for baseline and difference.

OBJECT TYPE	NAME	INSTANCES COUNT (BASELINE)	INSTANCES COUNT (DIFFERENCE)	MEMORY (BASELINE)	MEMORY (DIFFERENCE)
Procedure	Selectabc.p	100	0	70KB	0
OOABL	Windowabc.cls	50	0	50KB	0
OOABL	SampleTestClass.cls	30	0	45KB	0
OOABL	Sampletest.cls	45	0	45KB	0
OOABL	Test1.cls	60	0	37KB	0
OOABL	Test2.cls	75	0	35KB	0
OOABL	Test3.cls	66	0	30KB	0
OOABL	Test4.cls	75	0	29KB	0
Procedure	SelectTest.p	23	0	23KB	0
Procedure	Selectxyz.p	45	0	17KB	0
OOABL	Test2.cls	40	0	17KB	0
OOABL	Test3.cls	40	0	9KB	0
Procedure	Testxyz.p	40	0	7KB	0
Procedure	Testefg.p	40	0	6KB	0

Profiler Product Vision

Compare Snapshots

Compare snapshots taken at different times

- From a list of available snapshots, you can select any two for comparison and inspect the differences

Progress® OpenEdge Profiler

Comparison View

Profiles Import File

+ SNAPSHOTS Clear

- Snapshot3 ⋮
ABL_Application3/NetworkB
Size: 10MB Today, 2:00p.m
- Snapshot2 ⋮
ABL_Application3/NetworkB
Size: 20MB Today, 2:00p.m
- Snapshot1 ⋮
ABL_Application3/NetworkB
Size: 30MB Today, 2:00p.m
- Snapshot_Build2.1.1 ⋮
ABL_Application4/NetworkC
Size: 10MB Yesterday

+ RECORDINGS Clear

- Recording_Build2.1.1 ⋮
ABL_Application4/NetworkC
Size: 70MB 12/04/2022, 2:30p.m
- Recording_Build2.1.0 ⋮
ABL_Application4/NetworkC
Size: 80MB 12/04/2022, 10:30a.m

Snapshot3 (baseline) | Snapshot2

Application Object Types | Widget Pools | Garbage Collected Objects | Unmanaged Memory

Q Add filter...

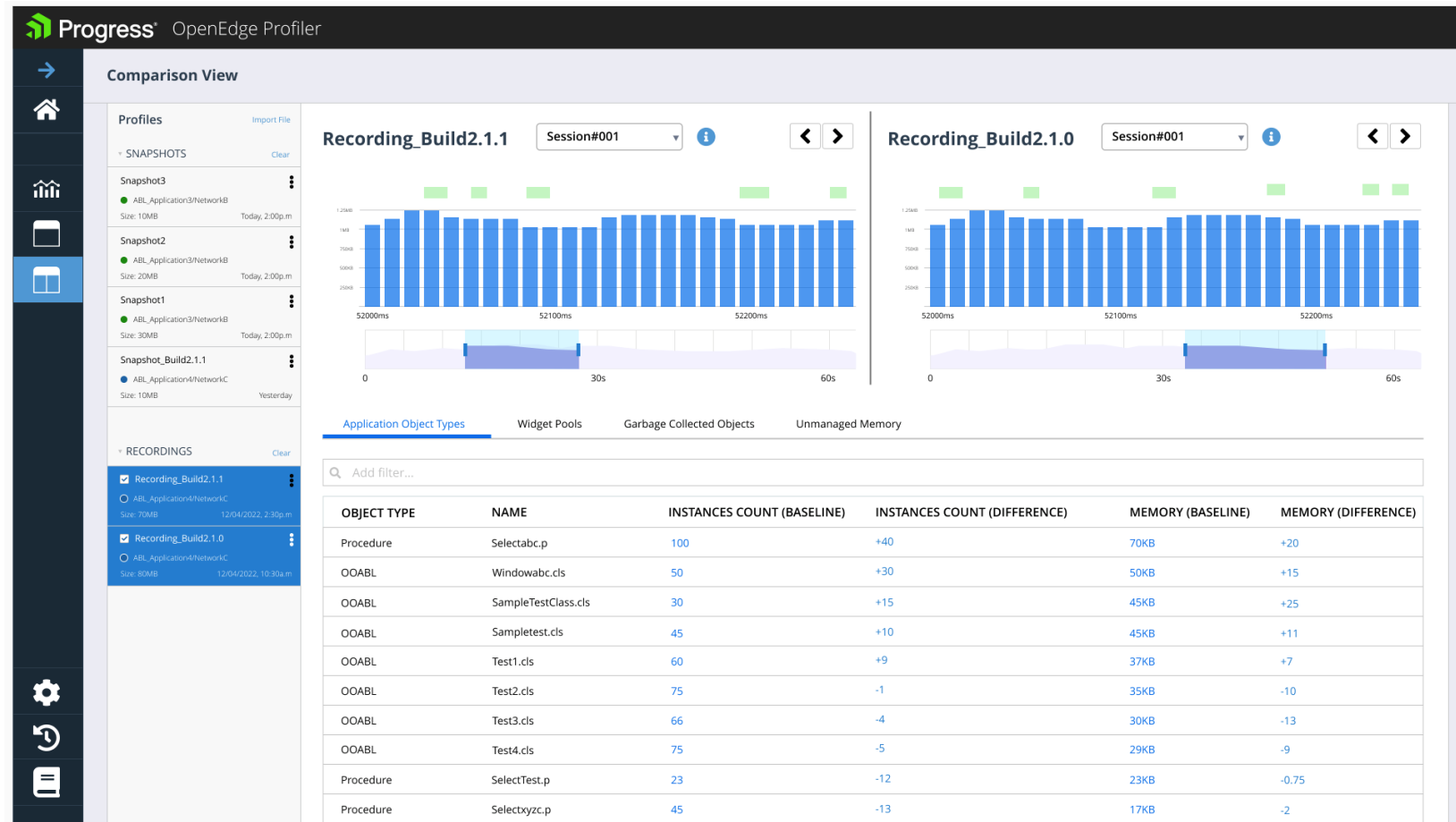
OBJECT TYPE	NAME	INSTANCES COUNT (BASELINE)	INSTANCES COUNT (DIFFERENCE)	MEMORY (BASELINE)	MEMORY (DIFFERENCE)
Procedure	Selectabc.p	100	+40	70KB	+20
OOABL	Windowabc.cls	50	+30	50KB	+15
OOABL	SampleTestClass.cls	30	+15	45KB	+25
OOABL	Sampletest.cls	45	+10	45KB	+11
OOABL	Test1.cls	60	+9	37KB	+7
OOABL	Test2.cls	75	-1	35KB	-10
OOABL	Test3.cls	66	-4	30KB	-13
OOABL	Test4.cls	75	-5	29KB	-9
Procedure	SelectTest.p	23	-12	23KB	-0.75
Procedure	Selectxyz.p	45	-13	17KB	-2
OOABL	Test2.cls	40	-15	17KB	-9
OOABL	Test3.cls	40	-20	9KB	-2
Procedure	Testxyz.p	40	-15	7KB	-1.6
Procedure	Testefg.p	40	-20	6KB	-20

Profiler Product Vision

Compare Recordings

Compare part of recordings taken at different times

- From a list of available recordings, you can select the range of time for any two to compare and inspect the differences



Customer Validation Program

Join the CVP!

OpenEdge Customer Validation Program

Actively influence the developer experience and future enhancements of Progress OpenEdge!

Get Access to:

Roadmap surveys
Usability reviews
Pre-release software

Virtual open houses
Quarterly objectives
Sprint reviews

progress.com/openedge/customer-validation-program



Questions (and a few FAQs)

Questions? (And a few FAQs)

Q. Will the memory profiler work with any OpenEdge release?

A. No. It will only work starting with the release the capability is offered in.

Q. What is the overhead of using the memory profiler?

A. With what is built out so far, we found the overhead to be less than 9% CPU overhead and 7% memory overhead.

Q. Will the profiler tool also have CPU profiling capability?

A. Yes, we intend to include CPU profiling but not in the initial release.

Q. Will the format of the memory profiling data be available publicly?

A. Yes

Questions? (And a few FAQs)

Q. Will the memory profiler also profile memory allocations of the .NET ABL objects?

A. We will try to do this, but no promises. The AVM can monitor memory use by .NET, but it can't track it in the same way.

News You Can Use



