Consultingwerk
software architecture and development

.NET on Linux. Why bother?

PUG Challenge 2026 Amsterdam

Peter Judge, Consultingwerk
peter.judge@consultingwerk.com

Peter Judge



Writing 4GL since 1996, working on a variety of frameworks and applications

Focusing on integration with OpenEdge applications, PASOE, software architecture and web technologies

Active member of the OpenEdge community and speaker at international conferences

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Consultingwerk
software architecture and development

Our Focus: Modernizing OpenEdge Applications



Modernization of Legacy OpenEdge Applications



Deep Technical Expertise



Global IT Partner with Local Presence



More than Consulting – We Deliver Tools & Solutions

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Six Days Before .NET 8 Goes Out of Support

4 Nov

This session
PUG Challenge Amsterdam

10 Nov

.NET 8 end of support:
the .NET certified for 13.0

Nov 2028

.NET 10 LTS
projected end of support

OpenEdge 13.0 is certified with .NET 8: the version question is not academic

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Remember Word and Excel With OpenXML?

PUG Challenge 2025

Word and Excel documents from ABL

Open XML SDK instead of Office and OCX

Runs on Linux, too

OpenEdge with .NET on Linux

No Office, no Windows server

It is just a .NET library

One NuGet package:

DocumentFormat.OpenXml

Many more where it came from

If Office documents work on Linux, what else does .NET bring?

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

What You'll Take Home

What works on Linux

OpenEdge 13.0 clients and PASOE

Startup parameters

Which .NET version

Where .NET beats ABL

Files and zip archives

Process execution

Certificates: X.509

And a few more

What it costs

Bridge overhead

Runtime deployment

Pitfalls to expect

If you have dismissed .NET on Linux, give it the next 55 minutes

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



What OpenEdge 13.0 Supports on Linux

What's in the .NET Box?

The runtime

Runs .NET code on Linux, Windows and macOS

Free download from Microsoft

LTS releases: .NET 8 and .NET 10

The class library

`System.IO`

`System.IO.Compression`

`System.Diagnostics`

`System.Security.Cryptography`

Ships with the runtime

NuGet packages

Everything else: Open XML SDK, Kafka, cloud SDKs

Added per application

.NET is not .NET Framework: 4.8 and the GUI stay on Windows

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Calling .NET Looks Like OOABL

hello-net.p

```
USING System.IO.* FROM ASSEMBLY.
USING System.Text.* FROM ASSEMBLY.

DEFINE VARIABLE aFiles AS "System.String[]" NO-UNDO.
DEFINE VARIABLE oSb    AS StringBuilder    NO-UNDO.
DEFINE VARIABLE oList  AS
  "System.Collections.Generic.List<CHARACTER>" NO-UNDO.

aFiles = Directory:GetFiles("/data", "*.xml").
oSb = NEW StringBuilder().
oSb:Append("Hello"):Append(" Linux").
MESSAGE oSb:Length oSb:ToString().

CATCH e AS System.Exception:
  MESSAGE e:Message.
END CATCH.
```

If you know OOABL

- `USING … FROM ASSEMBLY`
- Statics with a colon: `Directory:GetFiles`
- `NEW`, methods, properties
- Arrays and generics: type name in quotes
- .NET exceptions: `CATCH` them

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Since 12.8, ABL Talks to .NET on Linux



Character client

Interactive CHUI sessions

Same .NET code as on Windows



ABL batch client

Jobs, imports, cron

The natural home for plumbing



PASOE agent

Business logic behind REST, APSV and web handlers

OpenEdge 12.8 and 13.0: same syntax as on Windows

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Three Startup Parameters and One JSON File

```
$DLC/bin/_progres -b -p hello.p -clrnetcore -assemblies /opt/app/assemblies
```

-clrnetcore

Required on Linux, not combinable with `-noclr` (the default)

-assemblies

Folder with `assemblies.xml` and your own DLLs

-preloadCLR

Loads the CLR at startup instead of on first use

runtimeconfig.json in $DLC/bin

Decides which .NET version is loaded

Forget `-clrnetcore` and the first `NEW System.…` fails

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Bring Your Own Runtime



The .NET runtime

Not shipped with OpenEdge

Distro package or Microsoft's install script

Runtime is enough, SDK optional



Your assemblies

Not shipped either

Copy them into your assemblies folder

List them in `assemblies.xml`



NuGet native libraries

Some packages carry per-OS binaries

Extract them by hand

Progress example: Kafka client

OpenEdge brings the bridge. You bring everything that crosses it.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Which .NET? Read the Small Print

OpenEdge	.NET	What Progress says	Source
12.8 (initial)	.NET 6	First on Linux, out of support Nov 2024	What's New 12.8
12.8.12+	.NET 10	Certified on Windows and Linux	What's New 12.8
13.0	**.NET 8**	**Certified; .NET 8 EOL 10 Nov 2026**	**What's New 13.0**
13.0	.NET 10	Not supported	What's New 13.0

Progress KB article

This talk runs on 13.0 with .NET 8. Check the KB before you go live.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

The Linux File That Picks the .NET Version

```
$DLC/bin/Progress.clrbridge.netcore.runtimeconfig.json
```

As 13.0 installs it on Linux

```
{
  "runtimeOptions": {
    "tfm": "net6.0",
    "framework": {
      "name": "Microsoft.NETCore.App",
      "version": "6.0.0"
    }
  }
}
```

After the edit for .NET 8

```
{
  "runtimeOptions": {
    "tfm": "net8.0",
    "framework": {
      "name": "Microsoft.NETCore.App",
      "version": "8.0.0"
    }
  }
}
```

13.0 ships pointing at .NET 6. Editing this file is an install step, not a tweak.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Same ABL Code, Two Platforms

hello.p

```
USING System.Runtime.InteropServices.*
    FROM ASSEMBLY.

MESSAGE RuntimeInformation:OSDescription
        SKIP
        RuntimeInformation:FrameworkDescription
        SKIP
        RuntimeInformation:ProcessArchitecture
            :ToString().
```



You'll see

- Windows or Linux kernel string
- .NET 8.0.x on both
- X64 on both
- **No change to the source**

Non-GUI code moves between Windows and Linux without changes

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Where .NET Removes Complexity

Files, zip archives, processes and cryptography

File Handling: Walking a Directory Tree

ABL: a work queue in a temp-table

```
CREATE ttDir. ttDir.cPath = "/data/in".
DO WHILE CAN-FIND(FIRST ttDir):
  FIND FIRST ttDir.
  cDir = ttDir.cPath.
  DELETE ttDir.
  INPUT FROM OS-DIR(cDir).
  REPEAT:
    IMPORT cName cFull cAttr.
    IF cName = "." OR cName = ".."
      THEN NEXT.
    IF INDEX(cAttr, "D") > 0 THEN DO:
      CREATE ttDir. ttDir.cPath = cFull.
    END.
    ELSE IF cName MATCHES "*.xml" THEN
      oProcessor:ProcessFile(cFull).
  END.
  INPUT CLOSE.
END.
```

.NET: one call, filter and recursion included

```
USING System.IO.* FROM ASSEMBLY.
aFiles = Directory:GetFiles(
           "/data/in", "*.xml",
           SearchOption:AllDirectories).
DO i = 0 TO aFiles:Length - 1:
  oProcessor:ProcessFile(
      aFiles:GetValue(i):ToString()).
END.
```

Both sides call the same (fake) class

```
CLASS FileProcessor:
  METHOD PUBLIC VOID ProcessFile
      (pcFile AS CHARACTER):
    /* does stuff */
  END METHOD.
END CLASS.
```

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Zip Archives: Create, Change, List, Extract

.NET from ABL: System.IO.Compression

```
USING System.IO.Compression.* FROM ASSEMBLY.
/* a folder in, a folder out */
ZipFile:CreateFromDirectory("/data/in", "/tmp/in.zip").
ZipFile:ExtractToDirectory("/tmp/in.zip", "/tmp/out").
/* add an entry */
oZip = ZipFile:Open("/tmp/in.zip", ZipArchiveMode:Update).
ZipFileExtensions:CreateEntryFromFile(oZip,
    "/data/readme.txt", "docs/readme.txt").
oZip:Dispose().
/* list it: open again for reading */
oZip = ZipFile:Open("/tmp/in.zip", ZipArchiveMode:Read).
DO i = 0 TO oZip:Entries:Count - 1:
  MESSAGE oZip:Entries[i]:FullName oZip:Entries[i]:Length.
END.
oZip:Dispose().
```



ABL today

- Shell out to `zip` and `unzip`
- Is zip installed on the server?
- Listing: `unzip -l`, then parse
- Pure ABL zip: portable, but slow, small archives only

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 1: Files and Zip





Before

- OS-DIR work queue
- Filter by hand
- Zip via the shell

```
» Code:
aFiles = Directory:GetFiles(
  cRoot, "*.xml",
  SearchOption:AllDirectories).
MESSAGE aFiles:Length "files".

ZipFile:CreateFromDirectory(
  cRoot, "/tmp/in.zip").
```



You'll see

- File count from both programs
- Elapsed time side by side
- Entries of `in.zip`

One call replaces a work queue, one object replaces the shell

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Process Execution: OS-COMMAND Is a Blunt Tool

ABL: one stream, everything merged

```
INPUT THROUGH
  VALUE("git status --porcelain 2>&1").
REPEAT:
  IMPORT UNFORMATTED cLine.
  cOut = cOut + cLine + "~n".
END.
INPUT CLOSE.

/* exit code?  only via echo $? */
/* stderr?     merged into stdout */
/* timeout?    none               */
/* quoting?    your problem       */
```

.NET: exit code, stdout, stderr, timeout

```
oPsi = NEW ProcessStartInfo("git").
oPsi:ArgumentList:Add("status").
oPsi:ArgumentList:Add("--porcelain").
oPsi:RedirectStandardOutput = TRUE.
oPsi:RedirectStandardError  = TRUE.
oPsi:UseShellExecute        = FALSE.
oProc = Process:Start(oPsi).
oOut  = oProc:StandardOutput:ReadToEndAsync().
cErr  = oProc:StandardError:ReadToEnd().
cOut  = oOut:Result.
IF NOT oProc:WaitForExit(5000)
  THEN oProc:Kill().
iExit = oProc:ExitCode.
```

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

OS-COMMAND: What the Knowledge Base Says

No exit code

`OS-ERROR` is always 0

KB 000139689, 000145783

No timeout

The PASOE session time limit cannot stop it

KB 000228999

Slow on PASOE

Every call forks a big agent process

KB 000207833

Quoting is yours

Escape and whitelist every argument

KB 000172614, 000262145

Whose shell?

`PATH` from the owner's login scripts

KB 000228709, 000206419

Hard to stop

A hung child blocks the session

KB 000129692, 000254758

Process.Start avoids the shell and the fork slowdown. In PASOE it can't see the exit code.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

OS-COMMAND Slows Down as the Agent Grows

PASOE agent memory	`OS-COMMAND`	`Process:Start`	Ratio
~75 MB	427	181	2×
~1.1 GB	1,503	164	9×
~2.1 GB	**2,585**	**184**	**14×**

100 × `/bin/echo` from one PASOE agent, ms, median of 3 runs. Agent grown with a MEMPTR. OpenEdge 13.0, .NET 8.0.33, Ubuntu 26.04 (WSL).

Process:Start read stdout to the end: in a PASOE agent it cannot see the exit code.

OS-COMMAND slows down as the agent grows. Process.Start doesn't.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 2: Running a Process



Before

- `INPUT THROUGH`
- Output and errors in one blob
- No exit code

```
» Code:
oPsi = NEW ProcessStartInfo("ls").
oPsi:ArgumentList:Add("/data").
oPsi:ArgumentList:Add("/nope").
/* redirect stdout + stderr */
oProc = Process:Start(oPsi).
oProc:WaitForExit().
MESSAGE "exit:" oProc:ExitCode
  SKIP "err:" cErr.
```

You'll see

- Listing of /data on stdout
- Error for /nope on stderr
- Exit code 2

Exit code, stdout and stderr: three facts instead of one blob

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Security: Where ABL Stops

ABL has

- `MESSAGE-DIGEST`: SHA-256, SHA-512, HMAC (no SHA-384)
- `ENCRYPT` / `DECRYPT`: AES (CBC, ECB, GCM)
- `GENERATE-RANDOM-KEY`
- Certificates: only `sslc` via the shell

.NET adds

- X.509: load, inspect, verify the chain
- RSA and ECDSA: sign and verify
- PBKDF2 password hashing

Typical case: a partner's certificate. Is it valid, who signed it, when does it expire?

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

X.509 Certificates: From sslc Output to an Object

ABL today: sslc and text parsing

```
OS-COMMAND SILENT VALUE(
  "sslc x509 -in partner.pem"
  + " -noout -subject -enddate"
  + " > /tmp/cert.txt").
INPUT FROM /tmp/cert.txt.
REPEAT:
  IMPORT UNFORMATTED cLine.
  IF cLine BEGINS "notAfter=" THEN
    cEnd = ENTRY(2, cLine, "=").
END.
INPUT CLOSE.
/* "Jun  1 12:00:00 2027 GMT" */
/* parse month names by hand */
OS-COMMAND SILENT VALUE(
  "sslc verify -CAfile ca.pem"
  + " partner.pem > /tmp/v.txt").
/* then look for ": OK" */
```

.NET: the certificate is an object

```
USING System.Security.Cryptography.X509Certificates.*
    FROM ASSEMBLY.

oCert = NEW X509Certificate2("partner.pem").
MESSAGE oCert:Subject SKIP oCert:Issuer SKIP
        oCert:NotAfter.          /* a DATETIME */

oChain = NEW X509Chain().
oChain:ChainPolicy:TrustMode =
    X509ChainTrustMode:CustomRootTrust.
oChain:ChainPolicy:CustomTrustStore:Add(
    NEW X509Certificate2("ca.pem")).
oChain:ChainPolicy:RevocationMode =
    X509RevocationMode:NoCheck.   /* demo only */
MESSAGE "valid:" oChain:Build(oCert).
```

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Live Demo 3: Certificates



Before

- Shell out to `sslc`
- Temp files and text parsing
- Expiry date as a string

```
» Code:
RUN check-cert.p (
    "partner.pem", "ca.pem").

/* signed by another CA */
RUN check-cert.p (
    "stranger.pem", "ca.pem").
```

You'll see

- Subject, issuer, expiry
- Days until expiry
- Valid: yes, then no, with the reason

No shell, no parsing. Where keys and keystores live: Friday, 1:45 PM, New York 3

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

And More: Regex, Math, Time Zones, NuGet

Regular expressions

`Regex:IsMatch(cMail, cPattern)`

ABL only has `MATCHES` and `BEGINS`

In depth: Sebastian Wandel, Fri 6, 10:15

Math

`Math:Sin`, `Math:Atan2`, `Math:PI`

Distance between two coordinates

Time zones

`TimeZoneInfo:FindSystemTimeZoneById`

IANA ids such as `Europe/Amsterdam`

The NuGet ecosystem

Kafka, PDF, Excel, cloud SDKs

Watch for native libraries

If C# developers have a one-liner for it, so do you

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Faster?

Measured on 13.0, not borrowed from 2023

How We'll Measure

Setup

WSL, Ubuntu 26.04, one PC

OpenEdge 13.0, batch client

.NET 8.0.33

Workloads

2023 pair: `Guid.Parse`, distance

ABL vs .NET: SHA-256, directory walk, process start, zip

First .NET call

Method

`ETIME`, one warm-up run

Median of five runs

Check values must match

Same machine, same input, both sides of the bridge

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Why Measure Again?

	PUG Challenge Americas 2023	PUG Challenge 2026
Speaker	Mike Fechner	Peter Judge
OpenEdge	12.7	13.0
Platform	Windows	Linux (WSL)
Compared	.NET Framework 4.8 vs .NET 6	ABL vs .NET 8
Guid.Parse, 500,000 calls	6 s → 5 s	1.7 s
Distance, 10,000 calls	2.4 s → 1.9 s	0.6 s

Same two workloads, about 3× faster. Different hardware, so no lab claim.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

The Numbers

Workload (ms, median of 5)	ABL	.NET 8	What it shows
Guid.Parse, 500,000 calls	–	1,668	Bridge cost: ~3 µs a call
Distance, 500,000 calls	–	29,818	13 crossings; one C# call 14× faster
SHA-256, 100,000 strings	**56**	**1,161**	**ABL 20× faster**
Directory walk, 10,000 files	**23**	**4**	**.NET 6× faster**
100 process starts	221	224	Tie: the OS does the work
Zip, 10,000 files	130 (shell)	171	Tie; pure ABL ~95 s
First .NET call (CLR start)	–	130–215	Once per session

One big call wins. Many tiny calls lose.

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.



Running It in Production

PASOE, deployment and the pitfalls

PASOE: Parameters Belong to the Agent

```
# $CATALINA_BASE/conf/openedge.properties, section [AppServer.SessMgr.<abl-app>]
agentStartupParam=... -clrnetcore -assemblies /opt/app/assemblies -preloadCLR
```

Startup parameters

`-clrnetcore` goes into the agent startup parameters

Assemblies folder

Deployed with the instance, like your r-code

Preload the CLR

Pay the start-up once per agent, not in the first request

Set DOTNET_ROOT for the agent, or the first .NET call fails with 20825

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Gotchas: It's Not Just Code



Install .NET on every host

Runtime on each Linux server and in each image

Not always an easy sell to Linux sysadmins

Latest patch required for support



Assemblies in the pipeline

Build, version and deploy DLLs with the r-code

`assemblies.xml` per release

Artifact repository, e.g. Nexus



New .NET major, new compile

Edit `runtimeconfig.json`

Swap assemblies, recompile the r-code

LTS: three years, then again

Treat .NET like any other dependency: install, version, patch, upgrade

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Pitfalls to Expect

Events: not delivered

Background-thread events fail (15740): poll

Native NuGet libs

Download, extract, list: by hand

Vote: Idea OPENEDGE-I-1474

assemblies.xml

Framework ones too: zip, Process, crypto

Case-sensitive paths

`/Data/In` is not `/data/in`

IDE and build, 12.8.7+: `-casesensitiveclasses`

Async APIs

No `await` in ABL: `Task:Wait(), :Result`

Span-based overloads

Span-only APIs: wrap them in C#

Know them before the first deployment, not after

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

When .NET Pays Off, and When It Doesn't

Task	Stay with ABL	Reach for .NET
Read or write a single file	`COPY-LOB`, streams	–
Walk directory trees	Work queue	`Directory`
Create, change, list zips	Shell out	`ZipFile`, `ZipArchive`
Run external processes	Fire and forget	Exit code, stderr, timeout (not in PASOE)
Hashes and HMAC	`MESSAGE-DIGEST`	Only for SHA-384
Signatures and certificates	`sslc` via the shell	X.509, RSA, ECDSA
Tiny calls in hot loops	ABL	Only as one coarse call

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Don’t Dismiss It, Use It Where It Counts

It is supported

Since 12.8, also in 13.0

Batch, character, PASOE

Same code as Windows

It is practical

Files and zip archives

Processes

Cryptography

The NuGet ecosystem

It has a price

Bridge calls

Runtime deployment

Recompile per major

Pick the tool per task: ABL for the logic, .NET for the plumbing

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Get the Code



Samples and benchmarks on GitHub

- The programs from today's demos
- Files, zip, processes, certificates
- Benchmark code and results
- Setup notes for WSL



github.com/consultingwerk/DotNetCore-on-ABL-Linux-Samples-

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

My Sessions at PUG Challenge Amsterdam

Session	Day	Time	Room
.NET on Linux. Why bother?	**Wed 4**	**3:00 – 4:00 PM**	**New York 3**
What are user-defined generic types and why should I use them?	Thu 5	10:15 – 11:15 AM	New York 3
Managing secrets in OE configurations	Fri 6	1:45 – 2:45 PM	New York 3

Highlighted: this session

All sessions: pugchallenge.org

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Other Sessions Worth Your Time

Session	Speaker	Day	Time	Room
Regular Expressions for ABL Developers	Sebastian Wandel	Fri 6	10:15 – 11:15 AM	New York 2

All sessions: pugchallenge.org

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.

Questions

© 2026 Consultingwerk Application Modernization Solutions Ltd. All rights reserved.


Consultingwerk
software architecture and development