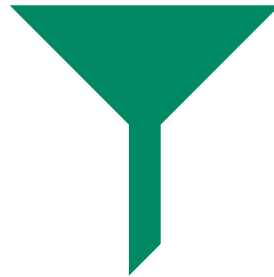# Agenda

- ABL Collections
  - Quick recap
  - Hash-Code() function
  - Hash based collections
- Dynamic Data Masking
- Protect Compiled R-Code
- Export and Import using dynamic buffers
- OpenTelemetry Tracing support
- Support for .NET on Linux
- Windows 32-bit support

OpenEdge ABL Collections

List

Delivered in 12.5

Sorted Set

Delivered in 12.6

HashMap

Delivered in 12.7

Progress®

HASH-CODE()

- Returns hash code as an integer for one or more arguments

- To support hash-based collection only

- Return value is consistent within a given ABL session

- Order of arguments can change the resulting hash code value

- Not replacement to ENCODE function, nor it is suitable for any cryptography tasks

- Any two values that return TRUE when compared with the EQ operator return the same hash code

HASH-CODE(arg [, arg] …)

Hash based Collections

- Hash based collections backed by hash table

- Hash table contains buckets for groups of objects

- Leverages Hash-Code function

HashMap

Progress®
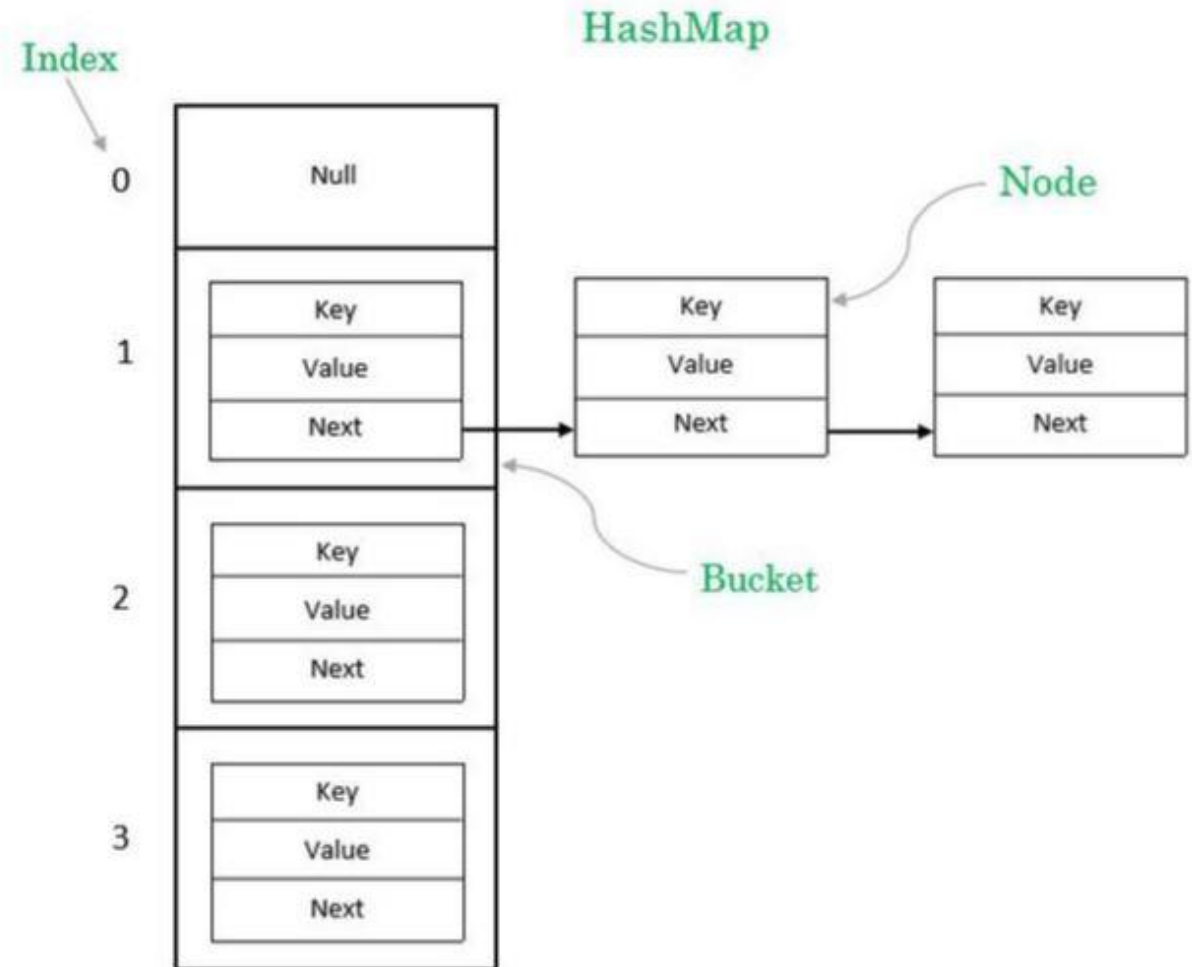
HashMap Collections

Collection of key-value pairs (referred to as a map)

Map contains unique keys

- Each key is associated with a value
- Values are not required to be unique
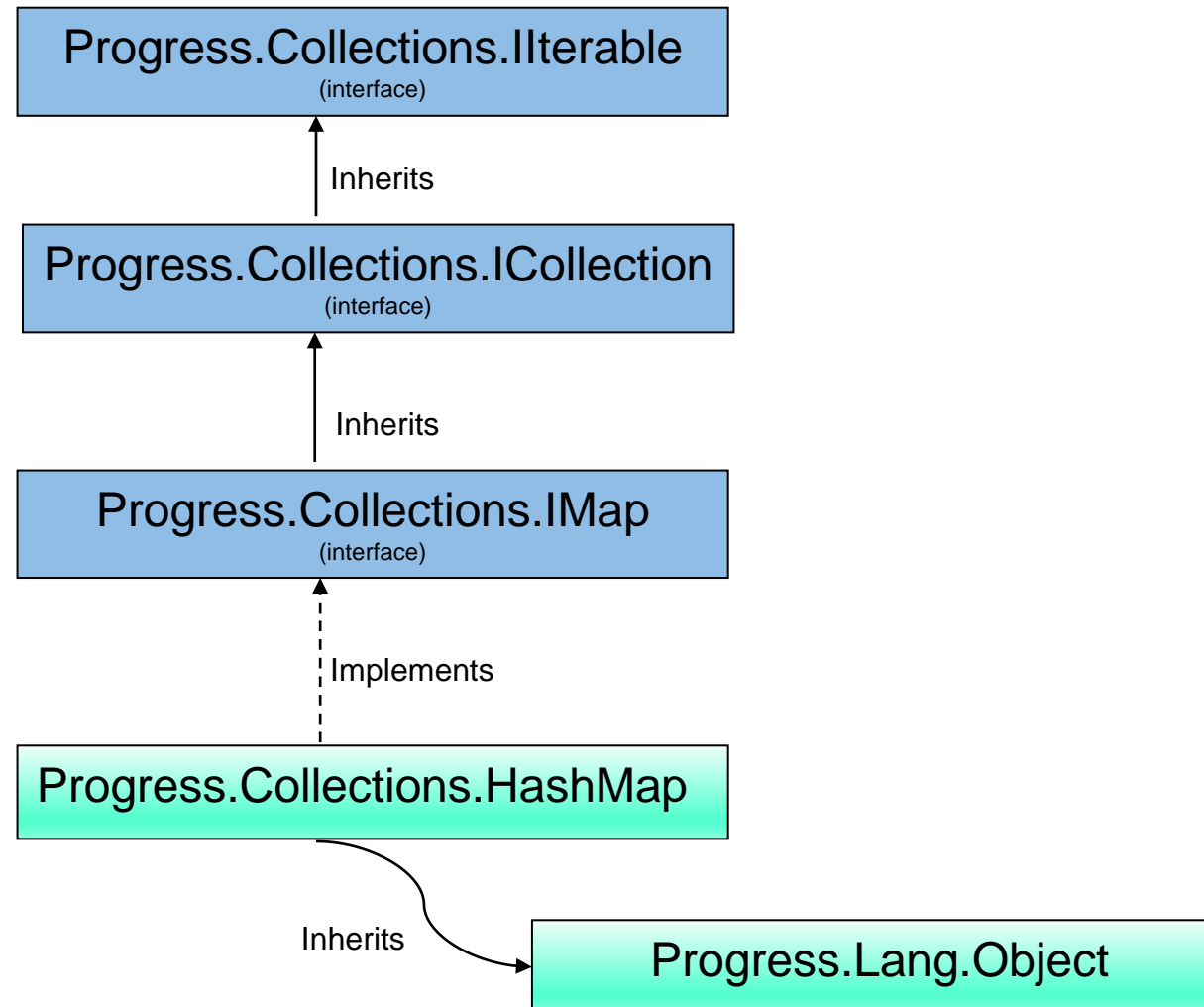
Ordering of elements is not important

Provides type safety

Most suited data structure for

- Better performance and
- Resource utilization

HashMap is serializable

Progress®

HashMap – Classes and Interfaces

```
        ┌──────────────────────────────────────┐
        │  Progress.Collections.IIterable      │
        │           (interface)                │
        └──────────────────────────────────────┘
                         ▲
                         │ Inherits
        ┌──────────────────────────────────────┐
        │  Progress.Collections.ICollection    │
        │           (interface)                │
        └──────────────────────────────────────┘
                         ▲
                         │ Inherits
        ┌──────────────────────────────────────┐
        │  Progress.Collections.IMap           │
        │           (interface)                │
        └──────────────────────────────────────┘
                         ▲
                         ┊ Implements
        ┌──────────────────────────────────────┐
        │  Progress.Collections.HashMap        │
        └──────────────────────────────────────┘
                         │ Inherits
                         ▼
                  ┌──────────────────────────────┐
                  │  Progress.Lang.Object        │
                  └──────────────────────────────┘
```

Progress.Collections.IHashable

Default comparer implementation
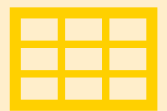
Requires HashCode() and Equals()
to be implemented

Progress.Collections.IEqualityComparer<T>
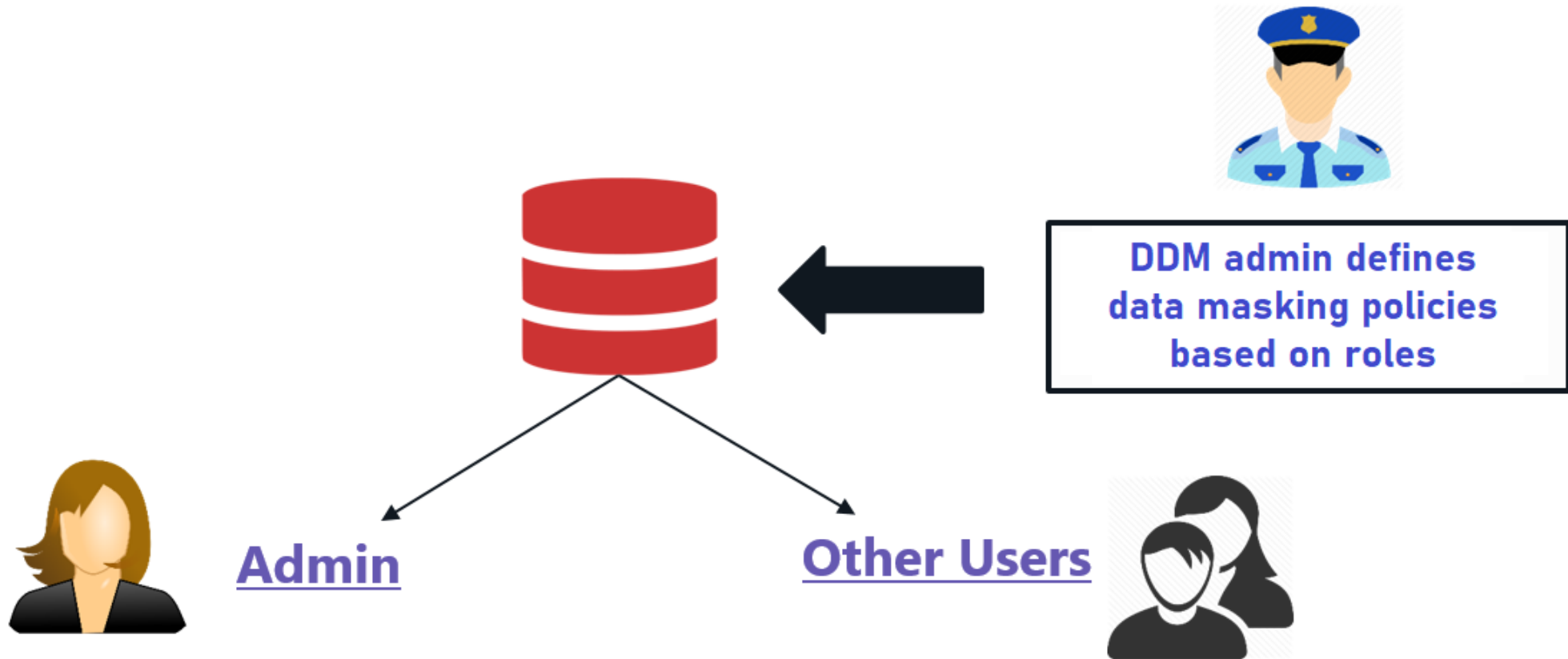
Where modifying class is not possible

Alternate implementations to
GetHashCode() and/or Equals()

Dynamic Data Masking (DDM)

Dynamic Data Masking (DDM)

DDM admin defines
data masking policies
based on roles

Admin

Other Users

First Name	Last Name	Age	email	SSN
Bob	Jones	48	bob@xms.com	123456789
Alis	Wonder	46	alis@xms.com	987654321

First Name	Last Name	Age	email	SSN
Bob	Jones	?	xxx@xxx.com	XXXXX789
Alis	Wonder	?	xxx@xxx.com	XXXXX321

AVM support for Dynamic Data Masking

A DDM admin will need to set up the following:

- Mask value

- Who is authorized to see the value without any masking.

The DDM configuration will be maintained via an ABL based tooling

Roles and tags
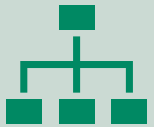
A DDM admin will grant the user-defined DDM role(s)

Associate those user-defined roles

Fields to be masked,

Linking them using 'authorization tags.'

Role will be associated with one or more tags

Allows those users to see the data, based on tags.

Mask Format

- To hide underlying data

- Transformed to same type as original

- Enforced to begin with either a 'D', 'N', 'L' or 'P', each followed by a colon

    D: Default mask

    N: Null value

    L: Literal value

    P: Partial masking

DDM Example - HR

```
FOR EACH employee:

    DISPLAY

        empnum

        firstname lastname

        startdate address city

        state1 postalcode

        birthdate ssn

        salary homephone

        homeemail SickDaysLeft.

END.
```

```
         Emp No:    1
     First Name: Kelly
      Last Name: Koberlein
     Start Date: 05/06/2015
        Address: 4500 Main Street
           City: Houston
          State: TX
    Postal Code: 75024
      Birthdate: 10/29/1986
            SSN: 700-12-1819
         Salary: $270,560
     Home Phone: (214) 234-4657
      HomeEmail: f_1_fakeaddr@gmail.com
 Sick Days Left: -1
```

DDM Example - Developer

```
FOR EACH employee:

    DISPLAY

        empnum

        firstname lastname

        startdate address city

        state1 postalcode

        birthdate ssn

        salary homephone

        homeemail SickDaysLeft.

END.
```

```
     Emp No:  1
  First Name: Kelly
   Last Name: Koberlein
  Start Date: 05/06/2015
     Address: *******
        City: *******
       State: TX
 Postal Code: 75024
   Birthdate: 01/01/1902
         SSN: ***-**-1819
      Salary: ?
  Home Phone: **************
   HomeEmail: f_1_**********l.com
Sick Days Left: -1
```

DDM Example – Authorized vs Unauthorized

Emp No: 1	Emp No: 1
First Name: Kelly	First Name: Kelly
Last Name: Koberlein	Last Name: Koberlein
Start Date: 05/06/2015	Start Date: 05/06/2015
Address: 4500 Main Street	Address: *******
City: Houston	City: *******
State: TX	State: TX
Postal Code: 75024	Postal Code: 75024
Birthdate: 10/29/1986	Birthdate: 01/01/1900
SSN: 700-12-1819	SSN: ***-**-1819
Salary: $270,560	Salary: ?
Home Phone: (214) 234-4657	Home Phone: **************
HomeEmail: f_1_fakeaddr@gmail.com	HomeEmail: f_1_**********l.com
Sick Days Left: 6	Sick Days Left: -1

DDM in action - Summary

Optionally establish one or more users as **secAdmin**

Enable database for DDM [enableddm]

Identify fields to be masked

For each field to be masked, Identify role – For unmasked information

Define a set of #DDM_See_ .. Authorization tags

For each of fields to be masked

- Assign authorization tag and
- Specify masking rule

Activate DDM in the database [activateddm]

Protect Compiled R-Code

Protecting Compiled Application Code

Hex Editor

Code
Injection

Compiled Application
Code

Database

Signed Package

Signed package for
compiled code
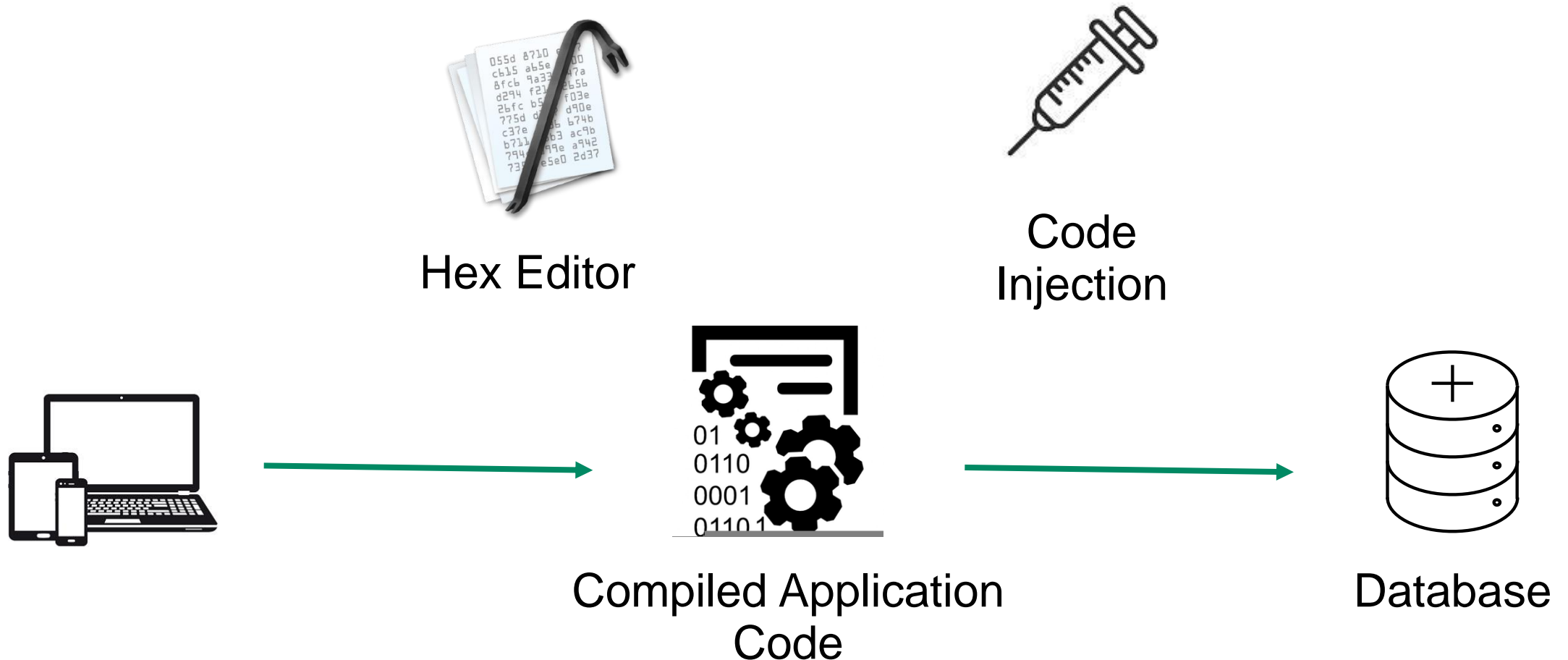
Database

Protect Compiled OpenEdge Code

Archive file with .apl extension

Based on Java JAR standard

Manifest file

Like procedure library but with better security

New tools – PROPACK and PROSIGN

Built-in Class: Progress.Archive.ArchiveInfo

Progress®

Export and Import using dynamic buffers

New methods

- BUFFER-EXPORT()

- BUFFER-IMPORT()

- BUFFER-EXPORT-FIELDS()

- BUFFER-IMPORT-FIELDS()

Progress®

OpenTelemetry Auto Tracing Support

OpenTelemetry - Introduction

- SDKs for telemetry data

  - Collect

  - Process

  - Export

- Supports telemetry data like

  - Logs

  - Metrics

  - Traces

- Vendor agnostic approach

OpenTelemetry

OpenTelemetry Collector

Middleware

Collect and manage data

Can be deployed on

- Host machine as agent
- Stand-alone service

Provides vendor neutral support to:

- Collect
- Process and
- Export telemetry data

Why do we need to implement?

- OpenEdge is part of larger distributed applications

- Enabling Cloud

- Assists in addressing:

  - Performance optimization

  - Troubleshooting

  - Problem isolation

- Customers are using monitoring tools like

  - Dynatrace,

  - NewRelic

Progress®

OpenEdge specific

ProOTel library — Wrapper library for OpenTelemetry

Auto tracing
- ABL Procedures
- Classes
- Transactions

Distributed tracing — Client and Server-side tracing

Other clients
- HTTP Client
- Java Open Client

Setup process

No changes to existing application code

Minimal setup

- AVM config file
- Collector config

Sample config

```json
{
    "OpenTelemetryConfiguration": {
        "exporters": {
            "otlp": {
                "http": [
                    {
                        "url": "http://localhost:25208/v1/traces",
                        .....
                    } ] } }
    },
    "OpenEdgeTelemetryConfiguration": {
        "trace_classes": Customer,Leads,Sales,Orders,
        "trace_procedures": sales_client.p,
        "trace_abl_transactions": false,
# The following two options are applicable to Distributed tracing.
        "trace_requires_parent": false,
        "trace_request_start": true
    }
}
```
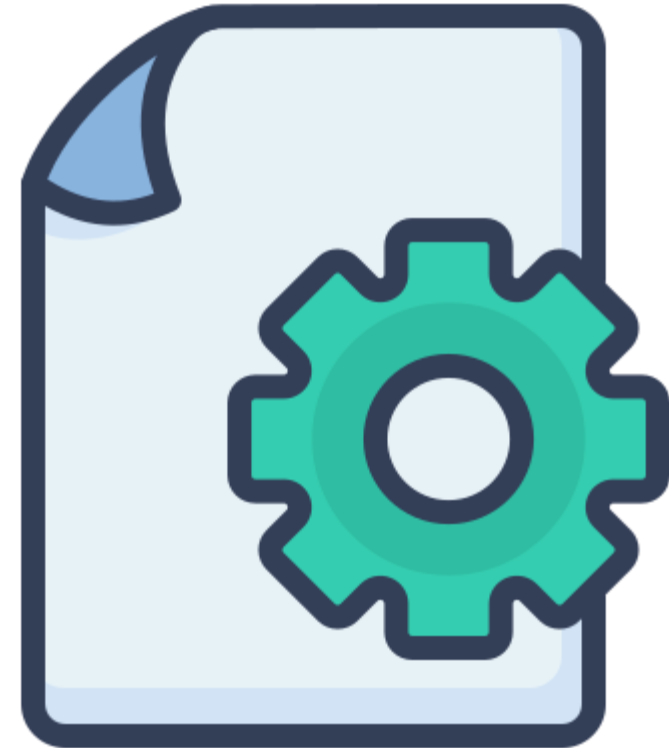
AVM: otelconfig

PASOE: otelConfigFile

Support for .NET on Linux

.NET on Linux

- Access .NET Classes from ABL on Linux
- Allows to run same app on both
  - Windows and
  - Linux

> Access to GUI functionality (say WinForms) on Linux is not supported

.NET Core
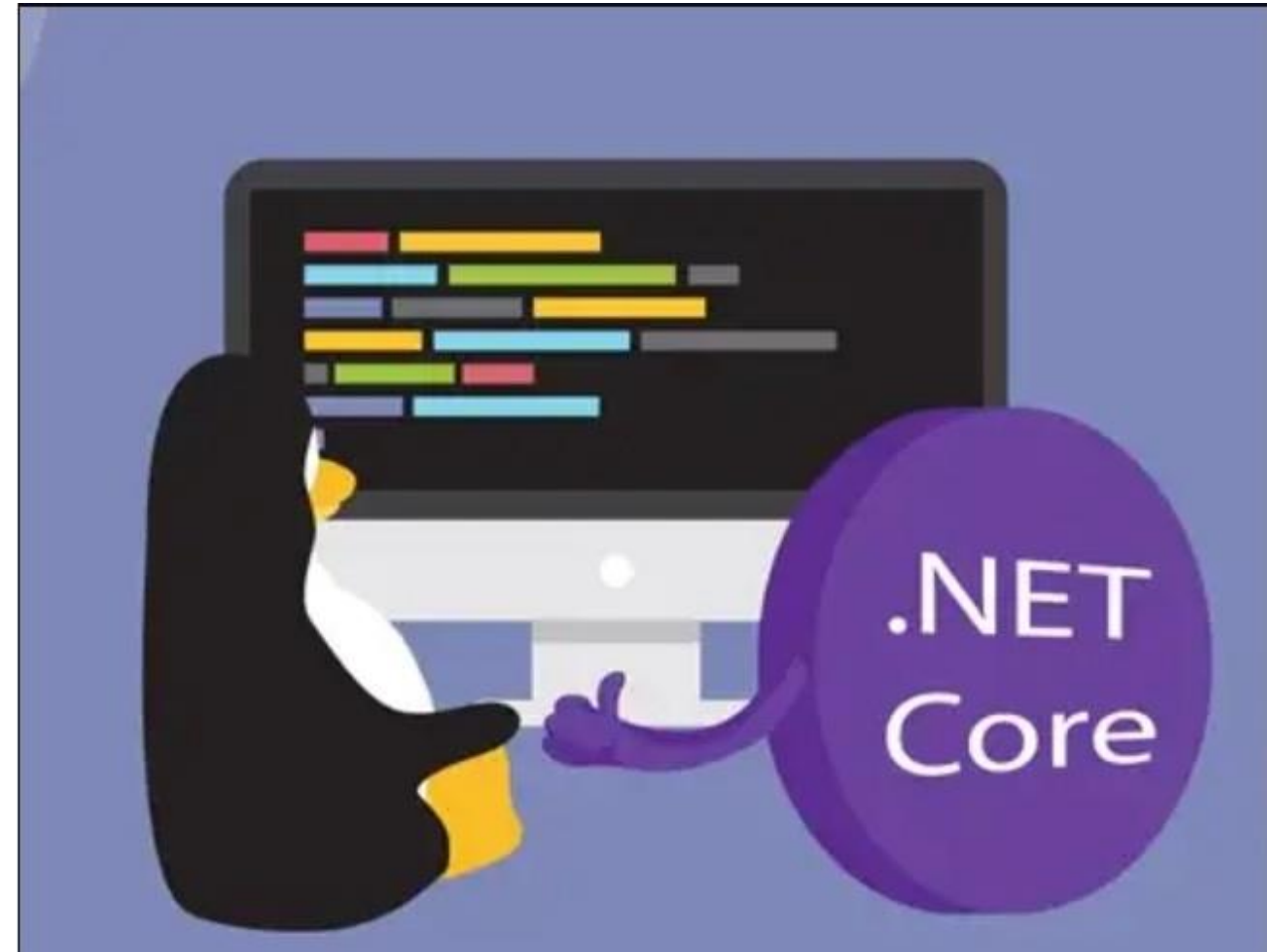
Supported clients

- Access .NET classes from clients:

  - Character client

  - ABL batch client

  - PASOE agent

Enables OpenEdge .NET cloud applications to be migrated to cost effective Linux platforms

How to enable this feature?

- Install .NET

- Use startup parameter
  - –clrnetcore

You cannot use -clrnetcore and No Common Language Runtime (-noclr) at the same time.
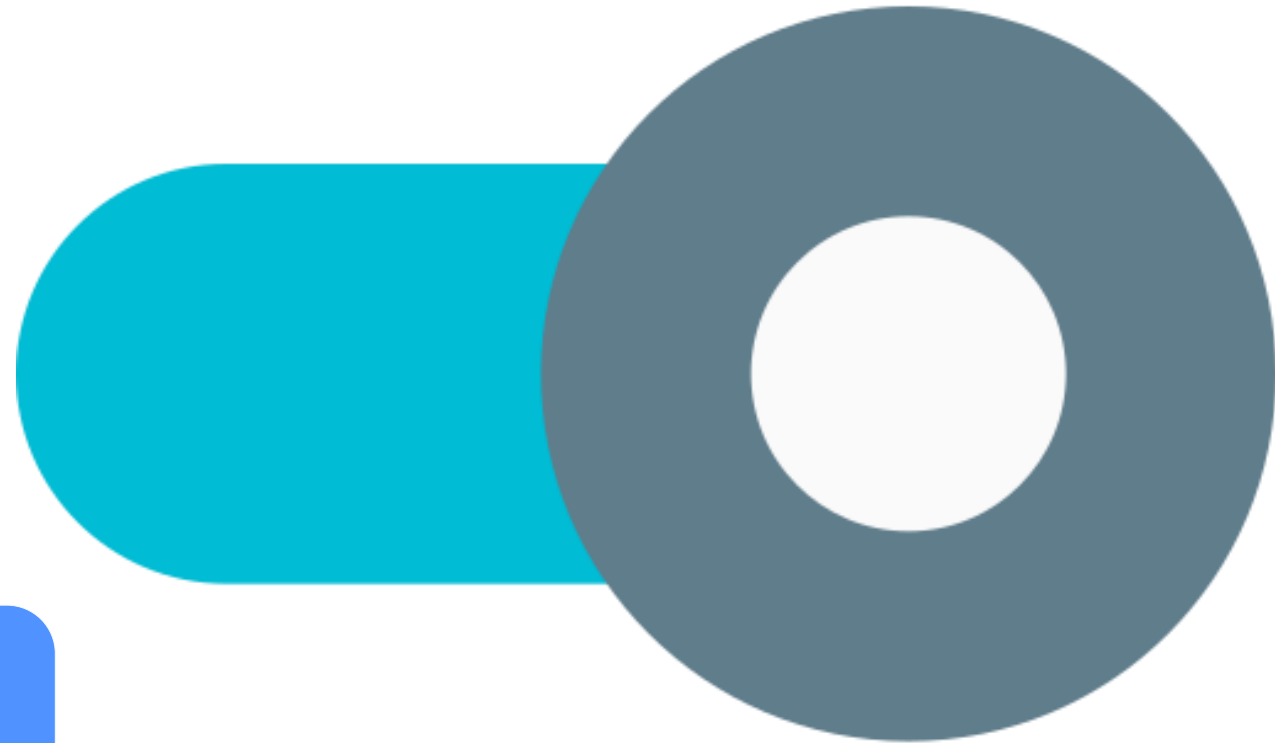
Progress®

Windows 32-bit support

Myths and facts about 32-bit Windows

Myth: You need a 32-bit operating system to run 32-bit applications.

✓ Fact: You can run 32-bit applications on a 64-bit operating system.

Myth: Progress dropped support for 32-bit applications in OE 12.0.

✓ Fact: 32-bit OE 12.x supports developing and running 32-bit applications.

✓ Fact: Progress has no plans to stop supporting the 32-bit Windows version of OpenEdge.

Myth: Microsoft dropped support for 32-bit applications in Windows 10.

✓ Fact: 64-bit versions of both Windows 10 and Windows 11 support running 32-bit applications.

✓ Fact: Microsoft has not announced plans to drop support for 32-bit applications.

> If this changes, Microsoft will announce it years in advance

---

Progress

32-bit OpenEdge applications on Windows
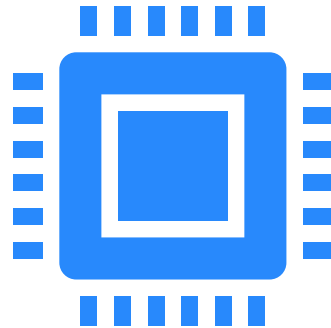
Problem Definition: Progress stopped offering the 32-bit version of OpenEdge Studio in 12.0.
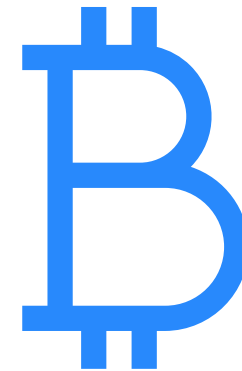
This left customers with no viable way to maintain applications which use 32-bit OCX controls and DLLs.

Progress®

Solution: 32-bit OpenEdge applications on Windows

✔ Progress added AppBuilder to 4GL Development System in 12.2.5 and 12.4.

Following will be added to 4GL Development System in upcoming release

Data Dictionary and Data Administration

No migration is required for 32-bit applications

Example: OCX controls or other components

Windows and OpenEdge both support 32-bit applications.

Solution

What did we learn?

HashMap Collections	OE - 12.7
Dynamic Data Masking	OE - 12.8
Protect Compiled R-Code	OE - 12.7
Export and Import using dynamic buffers	OE - 12.8
OpenTelemetry Tracing support	OE - 12.8
Support for .NET on Linux	OE - 12.8
Windows 32-bit support	OE - 12.2.x

Progress

Don't See Your Favorite Feature?

Submit and vote on ideas!

https://openedge.ideas.aha.io/

Join the CVP!

OpenEdge Customer Validation Program

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

Get Access to:

Roadmap surveys   Virtual open houses
Usability reviews   Quarterly objectives
Pre-release software   Sprint reviews

https://www.progress.com/openedge/customer-validation-program