Consultingwerk
software architecture and development

OpenEdge Legacy Application Modernization by Example

Mike Fechner
Director

Consultingwerk
software architecture and development

Consultingwerk Software Services Ltd.

Progress® Partner
PREMIER SERVICE DELIVERY
#ProgressProud

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiaries in UK and Romania
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration
- Experts in OpenEdge Application Modernization

Mike Fechner

- Director, Lead Modernization Architect and Product Manager of the SmartComponent Library and WinKit
- Specialized on object oriented design,software architecture, desktop user interfaces and web technologies
- 28 years of Progress experience (V5 … OE11)
- Active member of the OpenEdge community
- Frequent speaker at OpenEdge related conferences around the world

- Helps to protect your investment in your OpenEdge based application
- The framework is designed to modernize existing OpenEdge applications and to provide the foundation of new projects
- In the cloud and on premise
- UI flexibility – Desktop, Web & Mobile
- The architecture of the SmartComponent Library simplifies integration with future technologies and the implementation of new business requirements.

- **Introduction / Application Modernization**
- Modern OpenEdge Application Architecture
- ADM2 SDO migration
- TTY Upgrade editing migration
- ABL GUI migration
- OSIV3G Modernization example

Modernization Strategies

- Modernization of the whole application?
  - Going from ABL GUI to GUI for .NET or Web or Mobile
  - What is the *“final”* UI technology
  - GUI for .NET as an intermediate / integration with legacy GUI while the backend is rearchitected
- Or do we (first) add a few new features?
  - Mobile client for parts of the application
  - REST/REST(ful) interfaces for parts of the application

Quality of the application

- Are parts of the application reusable?
  - With no or little changes
  - Are major functional changes required?
  - Are major changes to the database structure required?

- Can parts of the application serve to describe the requirements
  - Legacy code review as part of the requirements definition
  - Is the existing source code the only (complete) description of the application functionality?

Skills of Development team

- New development process (let’s get agile)
- New tools (Progress Developer Studio, SCM, Unit Tests, Frontend tools)
- New architecture: Distributed
- New development languages
  - OOABL
  - html, JavaScript, TypeScript, rapidly changing
  - Desktop technologies

Modernization Examples in this presentation

- The modernization examples provided in this presentation demonstrate refactoring techniques based on simple examples
- These or similar techniques can be used for other types of ABL legacy applications
- Foundation for source code migration is always
  - understanding of existing code structure/architecture
  - understanding of target architecture
  - a concept
  - tools
  - experience
  - trial and error, or let’s call it a proof-of-concept

- Introduction / Application Modernization
- **Modern OpenEdge Application Architecture**
- ADM2 SDO migration
- TTY Upgrade editing migration
- ABL GUI migration
- OSIV3G Modernization example

OERA OpenEdge Reference Architecture

- Architecture blue print for service-oriented OpenEdge applications
- Initially released with OpenEdge 10.0 (15+ years)
- Primary goals at the time
  - AppServer enabling OpenEdge applications
  - Building non-monolithic OpenEdge applications
  - Supporting client flexibility
  - Providing guidance for use of the ProDataset
  - Providing guidance for use of OOABL (later, around OE10.1+)

OERA today

- Fast forward to 2015 …
- Modernization of OpenEdge applications more relevant than ever; especially since Telerik acquisition and demands for UI flexibility
- OEAA – OpenEdge Application Architecture, redefining the OERA
- OERA back on focus, foundation of the **CCS (common component specification)** project as a vehicle for community and Progress driven architecture-spec efforts
- More detailed specs, rather than just programming samples
- Specs that an application or framework could be certified against
- CCS starting to influence *“in-the-box”* features

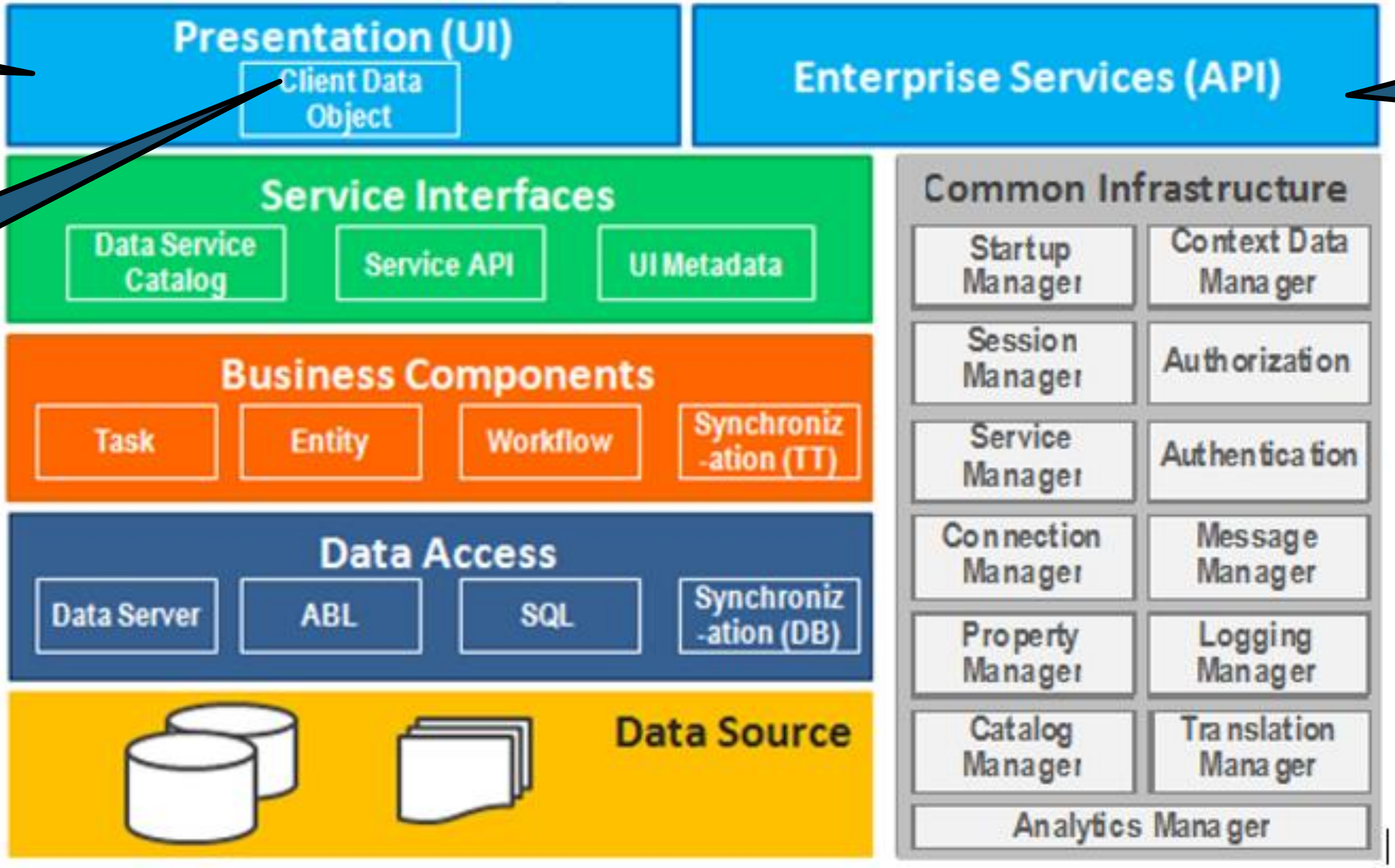
Business Entities

- Business Logic Component in the Business Service Layer
- Manages a set of database tables
  - Customer
  - Order/OrderLine/Item (read-only)
- CRUD actions (create, read, update, delete)
- Custom actions, verbs of the entity (PutCustomerOnCreditHold)
- Primary backend component for the JSDO
  - Kendo UI, Kendo UI Builder
  - NativeScript

The OpenEdge Application Architecture (OEAA)

Can be ABL GUI

That is the JSDO



Presentation (UI)

Client Data Object

Enterprise Services (API)

Service Interfaces

Data Service Catalog

Service API

UI Metadata

Business Components

Task

Entity

Workflow

Synchroniz -ation (TT)

Data Access

Data Server

ABL

SQL

Synchroniz -ation (DB)

Data Source

Common Infrastructure

Startup Manager

Context Data Manager

Session Manager

Authorization

Service Manager

Authentication

Connection Manager

Message Manager

Property Manager

Logging Manager

Catalog Manager

Translation Manager

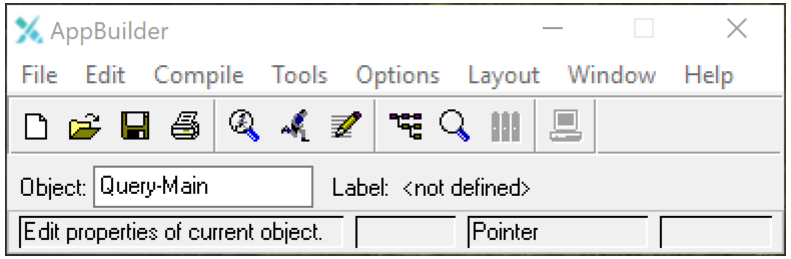
Analytics Manager

RESTful, SOAP, …

- Introduction / Application Modernization
- Modern OpenEdge Application Architecture
- **ADM2 SDO migration**
- TTY Upgrade editing migration
- ABL GUI migration
- OSIV3G Modernization example

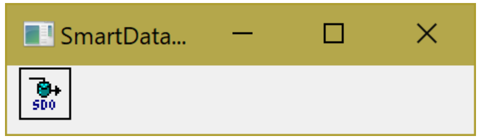
ADM2 SDO migration

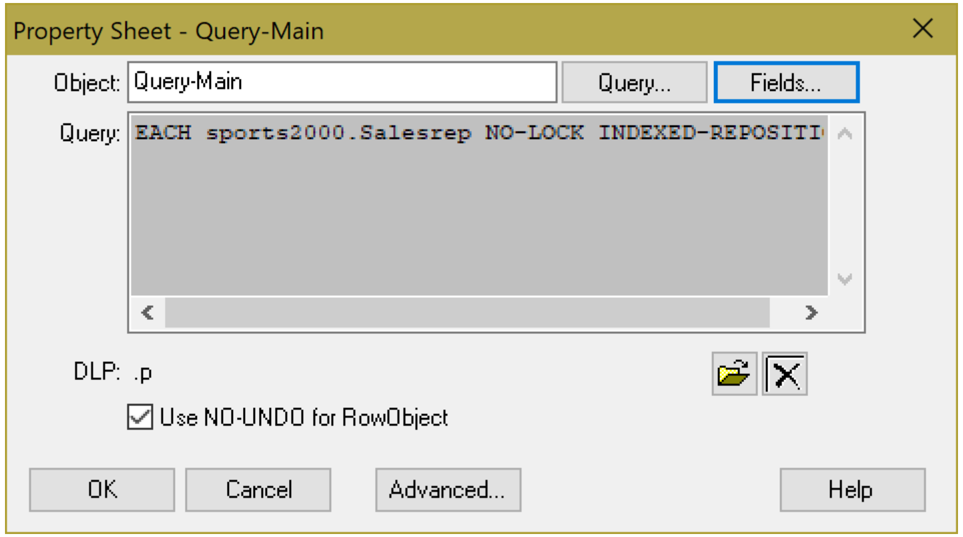
- SmartDataObjects (SDO’s) were introduced with Progress Version 9 and the ADM2
- SmartDataObjects have a similar responsibility within an application as a Business Entity
  - Centrally managing all read and update access to a database table
  - Based on temp-tables
  - Providing dedicated hooks for validation and calculated fields
  - Providing standards for change tracking and error reporting
  - Providing a central location for custom code that fits into the scope of the set of database tables



Object: Query-Main Label: <not defined>

Edit properties of current object. Pointer



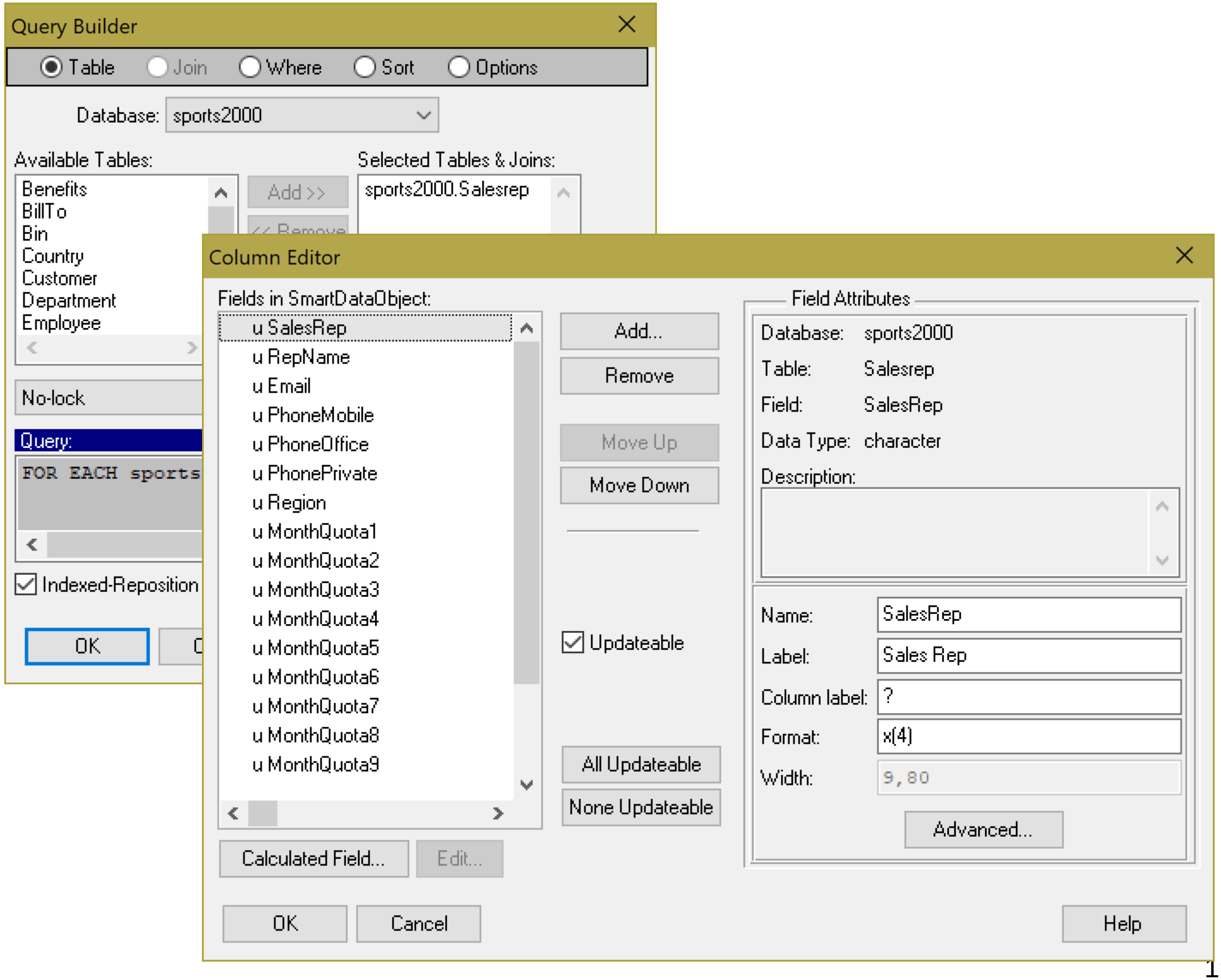


Object: Query-Main

Query: EACH sports2000.Salesrep NO-LOCK INDEXED-REPOSITI

DLP: .p

☑ Use NO-UNDO for RowObject



● Table ○ Join ○ Where ○ Sort ○ Options

Database: sports2000

Available Tables:
Benefits
BillTo
Bin
Country
Customer
Department
Employee

Selected Tables & Joins:
sports2000.Salesrep

No-lock

Query:

FOR EACH sports

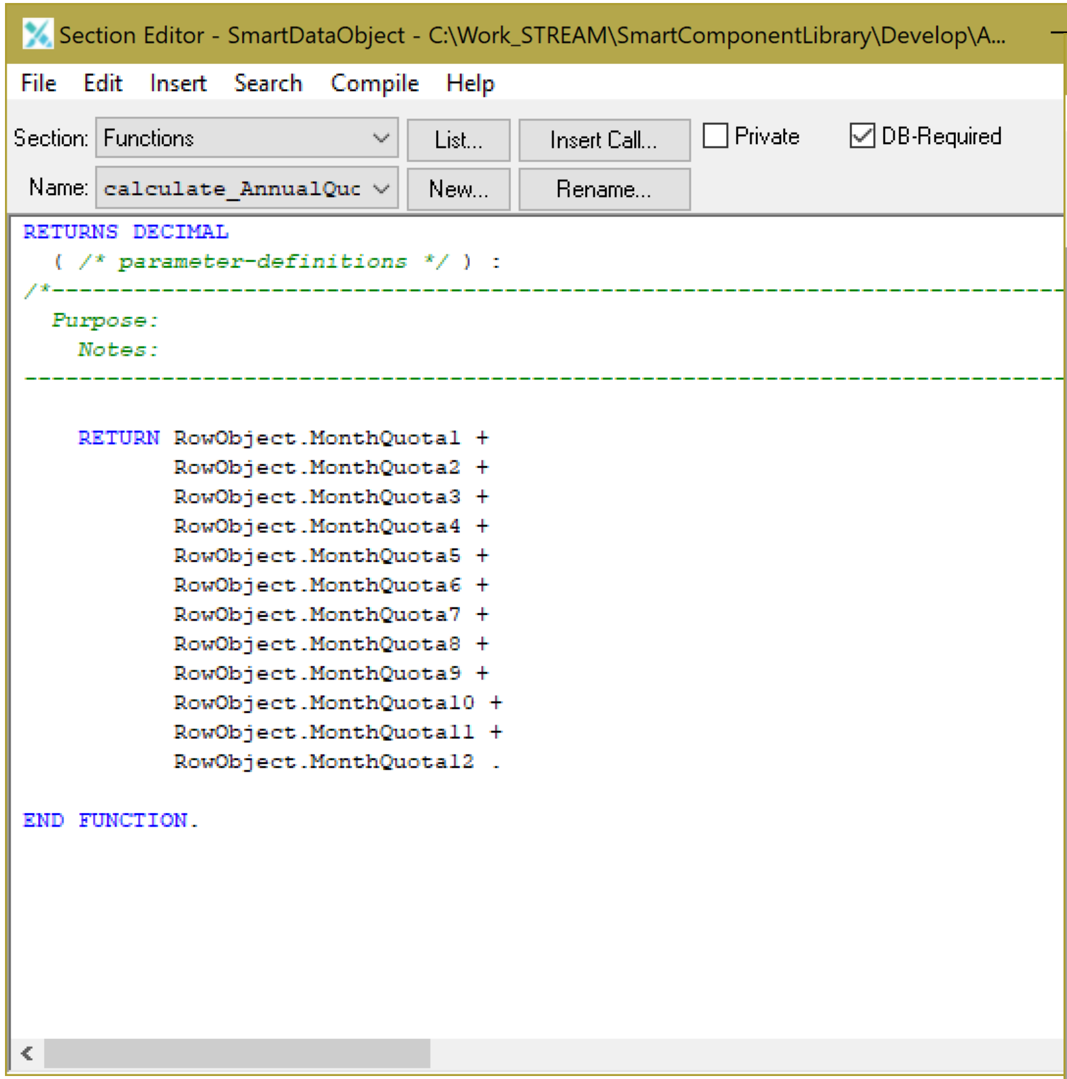
☑ Indexed-Reposition

Fields in SmartDataObject:
u SalesRep
u RepName
u Email
u PhoneMobile
u PhoneOffice
u PhonePrivate
u Region
u MonthQuota1
u MonthQuota2
u MonthQuota3
u MonthQuota4
u MonthQuota5
u MonthQuota6
u MonthQuota7
u MonthQuota8
u MonthQuota9

☑ Updateable

Field Attributes

Database:	sports2000
Table:	Salesrep
Field:	SalesRep
Data Type:	character
Description:	
Name:	SalesRep
Label:	Sales Rep
Column label:	?
Format:	x(4)
Width:	9,80



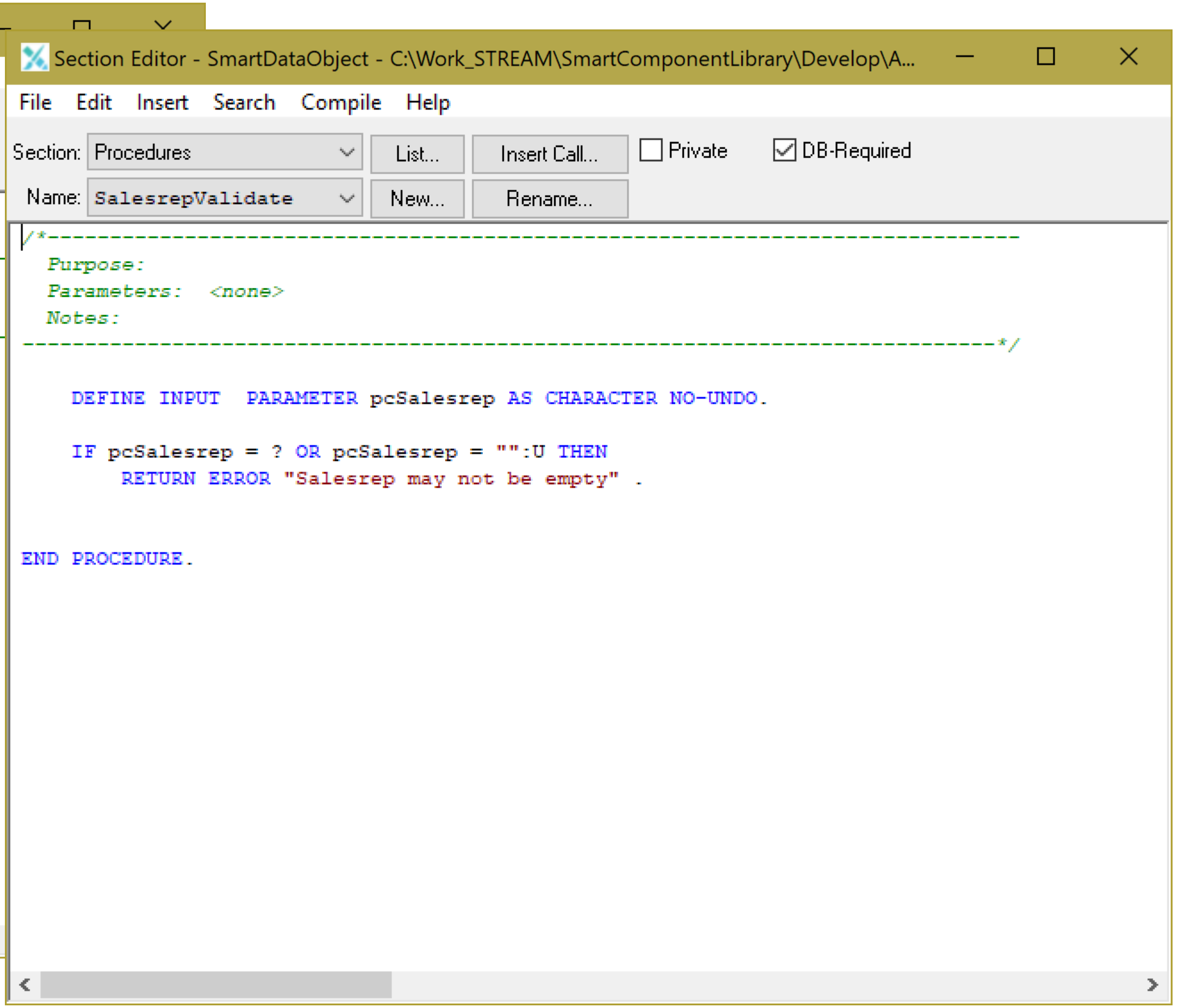
Section Editor - SmartDataObject - C:\Work_STREAM\SmartComponentLibrary\Develop\A...
File Edit Insert Search Compile Help
List...
Insert Call...
New...
Rename...

Section: Functions

Name: calculate_AnnualQuc

☐ Private ☑ DB-Required

```
RETURNS DECIMAL
  ( /* parameter-definitions */ ) :
/*------------------------------------------------------------------------------
  Purpose:
    Notes:
------------------------------------------------------------------------------

    RETURN RowObject.MonthQuota1 +
           RowObject.MonthQuota2 +
           RowObject.MonthQuota3 +
           RowObject.MonthQuota4 +
           RowObject.MonthQuota5 +
           RowObject.MonthQuota6 +
           RowObject.MonthQuota7 +
           RowObject.MonthQuota8 +
           RowObject.MonthQuota9 +
           RowObject.MonthQuota10 +
           RowObject.MonthQuota11 +
           RowObject.MonthQuota12 .

END FUNCTION.
```



Section Editor - SmartDataObject - C:\Work_STREAM\SmartComponentLibrary\Develop\A...
File Edit Insert Search Compile Help
List...
Insert Call...
New...
Rename...

Section: Procedures

Name: SalesrepValidate

☐ Private ☑ DB-Required

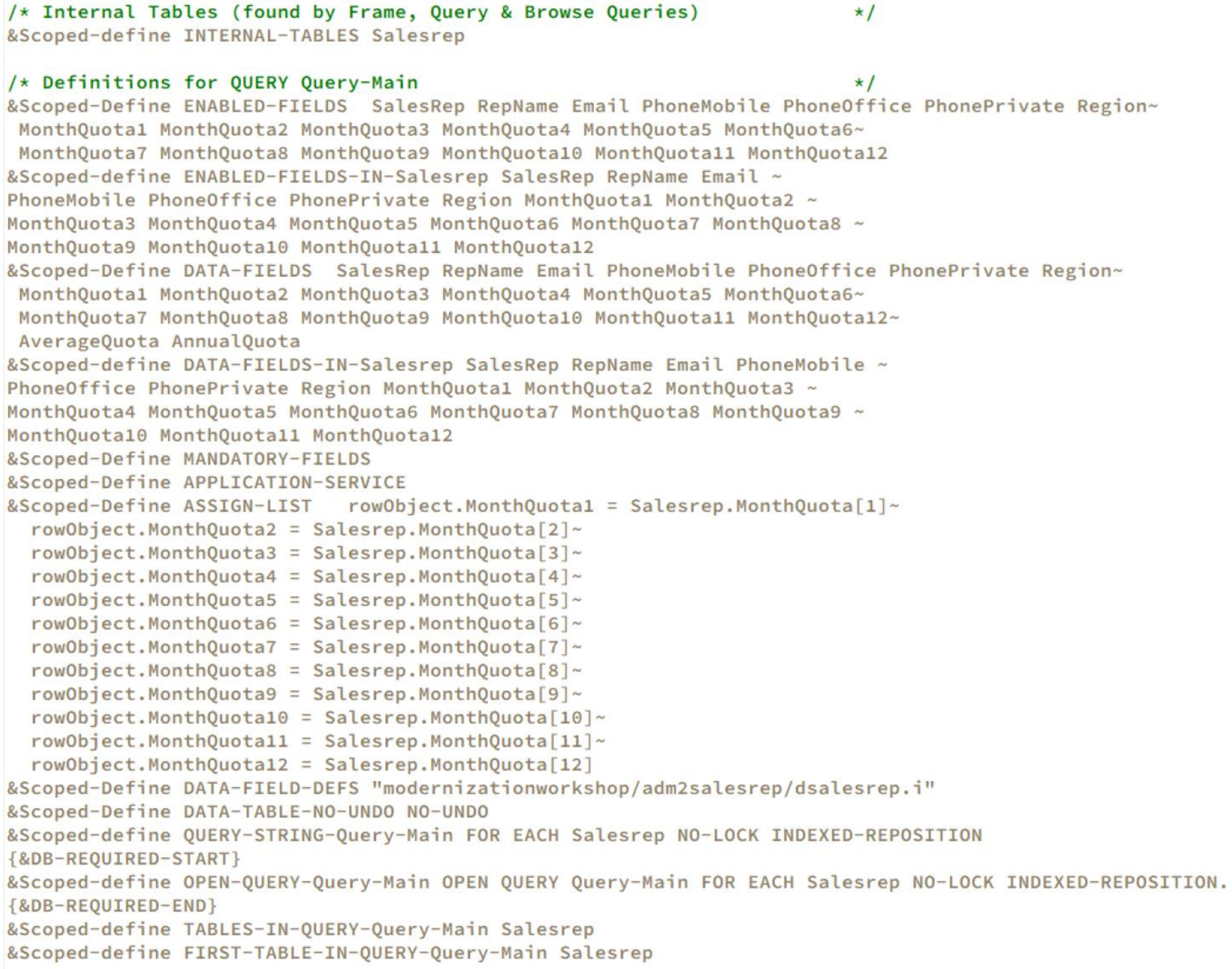
```
/*------------------------------------------------------------------------------
  Purpose:
  Parameters:  <none>
  Notes:
------------------------------------------------------------------------------*/

    DEFINE INPUT  PARAMETER pcSalesrep AS CHARACTER NO-UNDO.

    IF pcSalesrep = ? OR pcSalesrep = "":U THEN
        RETURN ERROR "Salesrep may not be empty" .


END PROCEDURE.
```

Reasons to migrate SDO’s to Business Entities

- Procedural nature
- Unclear separation between frontend and backend
- Complicated API when used from outside the ADM2
- Customization complicated, lots of code, understood only by a few developers
- Single table interface, Proprietary change tracking mechanism based on two temp-tables (a prototype of the ProDataset)
- AppBuilder tooling required for ADM2
- ProDataset better supported with modern tooling and UI

SDO migration

- Well defined source code structure
- Well defined patterns for internal procedures/functions
- Meta data defined in preprocessor directives
- SDO RowObject temp-table can serve as foundation for Business Entities



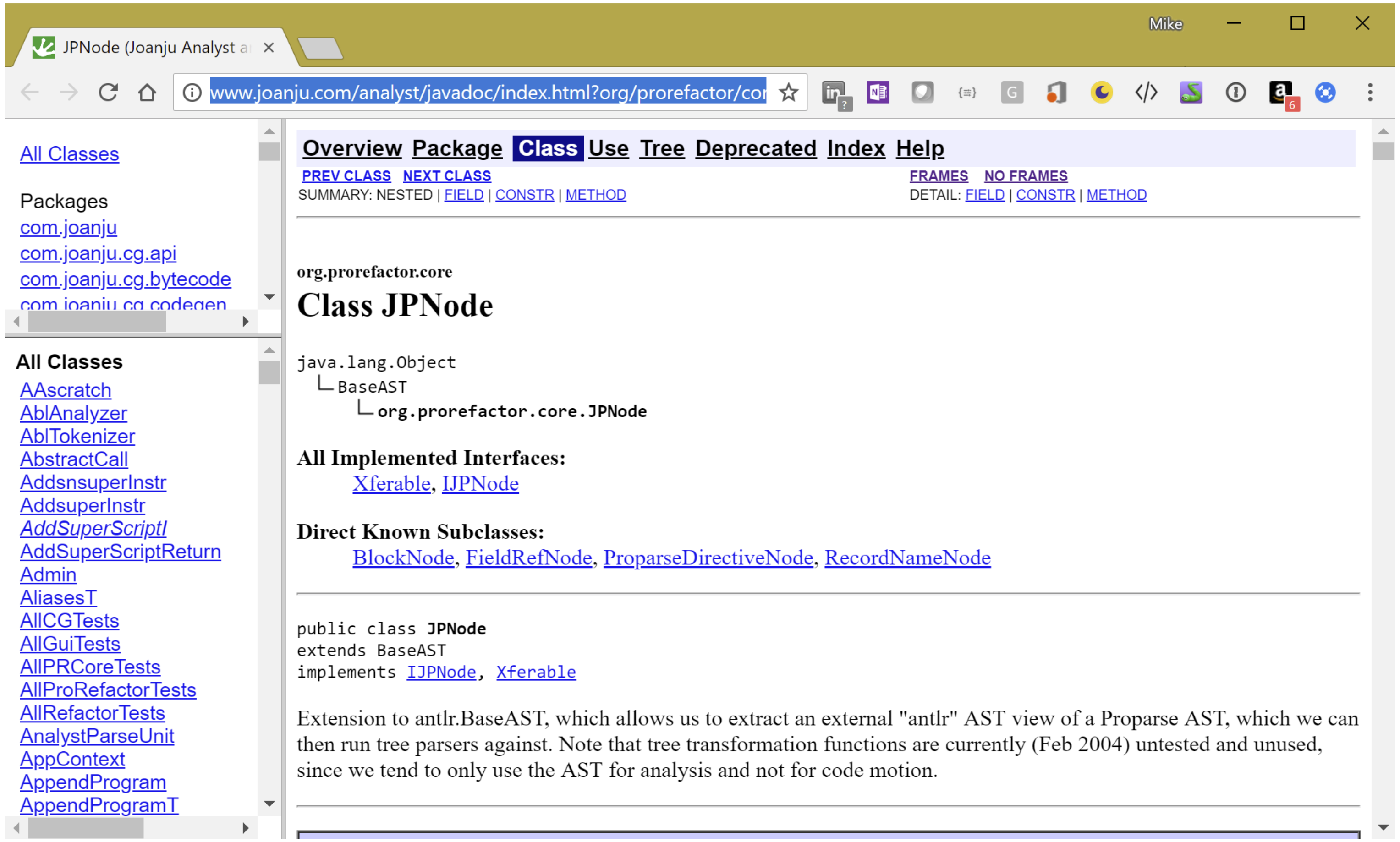
```
/* Internal Tables (found by Frame, Query & Browse Queries)              */
&Scoped-define INTERNAL-TABLES Salesrep

/* Definitions for QUERY Query-Main                                      */
&Scoped-Define ENABLED-FIELDS  SalesRep RepName Email PhoneMobile PhoneOffice PhonePrivate Region~
 MonthQuota1 MonthQuota2 MonthQuota3 MonthQuota4 MonthQuota5 MonthQuota6~
 MonthQuota7 MonthQuota8 MonthQuota9 MonthQuota10 MonthQuota11 MonthQuota12
&Scoped-define ENABLED-FIELDS-IN-Salesrep SalesRep RepName Email ~
PhoneMobile PhoneOffice PhonePrivate Region MonthQuota1 MonthQuota2 ~
MonthQuota3 MonthQuota4 MonthQuota5 MonthQuota6 MonthQuota7 MonthQuota8 ~
MonthQuota9 MonthQuota10 MonthQuota11 MonthQuota12
&Scoped-Define DATA-FIELDS  SalesRep RepName Email PhoneMobile PhoneOffice PhonePrivate Region~
 MonthQuota1 MonthQuota2 MonthQuota3 MonthQuota4 MonthQuota5 MonthQuota6~
 MonthQuota7 MonthQuota8 MonthQuota9 MonthQuota10 MonthQuota11 MonthQuota12~
 AverageQuota AnnualQuota
&Scoped-define DATA-FIELDS-IN-Salesrep SalesRep RepName Email PhoneMobile ~
PhoneOffice PhonePrivate Region MonthQuota1 MonthQuota2 MonthQuota3 ~
MonthQuota4 MonthQuota5 MonthQuota6 MonthQuota7 MonthQuota8 MonthQuota9 ~
MonthQuota10 MonthQuota11 MonthQuota12
&Scoped-Define MANDATORY-FIELDS
&Scoped-Define APPLICATION-SERVICE
&Scoped-Define ASSIGN-LIST   rowObject.MonthQuota1 = Salesrep.MonthQuota[1]~
  rowObject.MonthQuota2 = Salesrep.MonthQuota[2]~
  rowObject.MonthQuota3 = Salesrep.MonthQuota[3]~
  rowObject.MonthQuota4 = Salesrep.MonthQuota[4]~
  rowObject.MonthQuota5 = Salesrep.MonthQuota[5]~
  rowObject.MonthQuota6 = Salesrep.MonthQuota[6]~
  rowObject.MonthQuota7 = Salesrep.MonthQuota[7]~
  rowObject.MonthQuota8 = Salesrep.MonthQuota[8]~
  rowObject.MonthQuota9 = Salesrep.MonthQuota[9]~
  rowObject.MonthQuota10 = Salesrep.MonthQuota[10]~
  rowObject.MonthQuota11 = Salesrep.MonthQuota[11]~
  rowObject.MonthQuota12 = Salesrep.MonthQuota[12]
&Scoped-Define DATA-FIELD-DEFS "modernizationworkshop/adm2salesrep/dsalesrep.i"
&Scoped-Define DATA-TABLE-NO-UNDO NO-UNDO
&Scoped-define QUERY-STRING-Query-Main FOR EACH Salesrep NO-LOCK INDEXED-REPOSITION
{&DB-REQUIRED-START}
&Scoped-define OPEN-QUERY-Query-Main OPEN QUERY Query-Main FOR EACH Salesrep NO-LOCK INDEXED-REPOSITION.
{&DB-REQUIRED-END}
&Scoped-define TABLES-IN-QUERY-Query-Main Salesrep
&Scoped-define FIRST-TABLE-IN-QUERY-Query-Main Salesrep
```

Source code parsing using Proparse

- ABL syntax parser, abstract view on ABL source code, based on ANTLR
- Eliminates the need for text based source code analysis
  - Resolves issues with line-breaks, abbreviated keywords, mixed order of keywords
- Open source
  - github.com/oehive/proparse
  - github.com/consultingwerk/proparse
  - github.com/riverside-software/proparse
- Actively maintained in various forks, support for 11.7 ABL syntax

Proparse

- http://www.joanju.com/analyst/javadoc/index.html?org/prorefactor/core/JPNode.html


JPNode (Joanju Analyst a
Mike
www.joanju.com/analyst/javadoc/index.html?org/prorefactor/cor

All Classes

Packages
com.joanju
com.joanju.cg.api
com.joanju.cg.bytecode
com.joanju.cg.codegen

All Classes
AAscratch
AblAnalyzer
AblTokenizer
AbstractCall
AddsnsuperInstr
AddsuperInstr
AddSuperScriptI
AddSuperScriptReturn
Admin
AliasesT
AllCGTests
AllGuiTests
AllPRCoreTests
AllProRefactorTests
AllRefactorTests
AnalystParseUnit
AppContext
AppendProgram
AppendProgramT

Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

org.prorefactor.core

Class JPNode

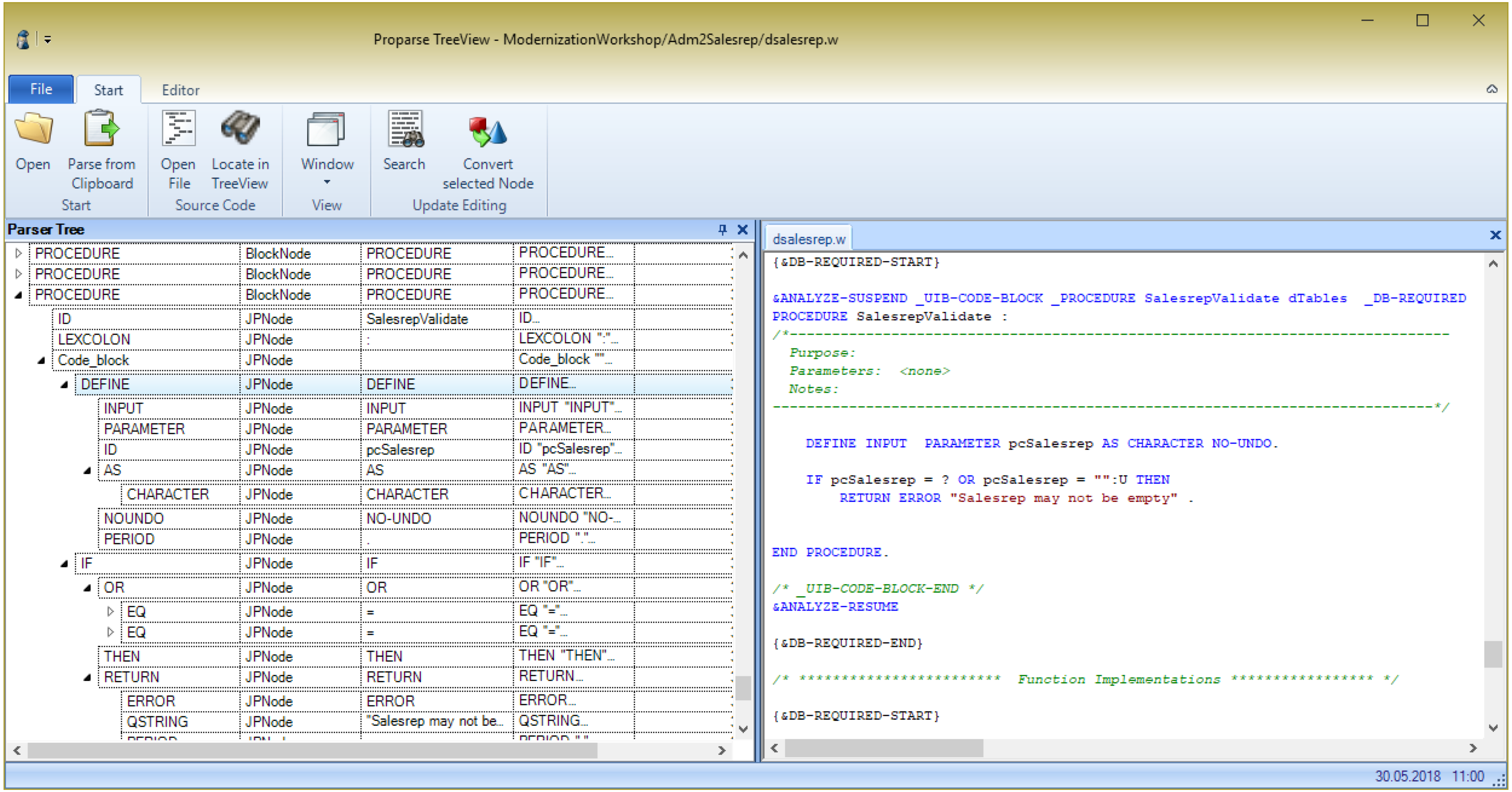
```
java.lang.Object
  └BaseAST
     └org.prorefactor.core.JPNode
```

All Implemented Interfaces:
Xferable, IJPNode

Direct Known Subclasses:
BlockNode, FieldRefNode, ProparseDirectiveNode, RecordNameNode

```
public class JPNode
extends BaseAST
implements IJPNode, Xferable
```

Extension to antlr.BaseAST, which allows us to extract an external "antlr" AST view of a Proparse AST, which we can then run tree parsers against. Note that tree transformation functions are currently (Feb 2004) untested and unused, since we tend to only use the AST for analysis and not for code motion.



PROCEDURE	BlockNode	PROCEDURE	PROCEDURE...
PROCEDURE	BlockNode	PROCEDURE	PROCEDURE...
PROCEDURE	BlockNode	PROCEDURE	PROCEDURE...
ID	JPNode	SalesrepValidate	ID...
LEXCOLON	JPNode	:	LEXCOLON ":"...
Code_block	JPNode		Code_block ""...
DEFINE	JPNode	DEFINE	DEFINE...
INPUT	JPNode	INPUT	INPUT "INPUT"...
PARAMETER	JPNode	PARAMETER	PARAMETER...
ID	JPNode	pcSalesrep	ID "pcSalesrep"...
AS	JPNode	AS	AS "AS"...
CHARACTER	JPNode	CHARACTER	CHARACTER...
NOUNDO	JPNode	NO-UNDO	NOUNDO "NO-...
PERIOD	JPNode	.	PERIOD "."...
IF	JPNode	IF	IF "IF"...
OR	JPNode	OR	OR "OR"...
EQ	JPNode	=	EQ "="...
EQ	JPNode	=	EQ "="...
THEN	JPNode	THEN	THEN "THEN"...
RETURN	JPNode	RETURN	RETURN...
ERROR	JPNode	ERROR	ERROR...
QSTRING	JPNode	"Salesrep may not be...	QSTRING...

```
{&DB-REQUIRED-START}

&ANALYZE-SUSPEND _UIB-CODE-BLOCK _PROCEDURE SalesrepValidate dTables  _DB-REQUIRED
PROCEDURE SalesrepValidate :
/*------------------------------------------------------------------------------
  Purpose:
  Parameters:  <none>
  Notes:
------------------------------------------------------------------------------*/

    DEFINE INPUT  PARAMETER pcSalesrep AS CHARACTER NO-UNDO.

    IF pcSalesrep = ? OR pcSalesrep = "":U THEN
        RETURN ERROR "Salesrep may not be empty" .


END PROCEDURE.

/* _UIB-CODE-BLOCK-END */
&ANALYZE-RESUME

{&DB-REQUIRED-END}

/* ************************  Function Implementations ***************** */

{&DB-REQUIRED-START}
```

SDO to Business Entity Migration

- SDO structure imported into SmartComponent Library Business Entity Designer
- Functionality implemented as a plugin to the tool
  - Not relevant for all users of the Business Entity Designer, can be disabled
  - Allows for easy customization in fork of the plugin
- Wizard supports changes to the SDO structure, e.g. adding/removing/renaming fields of the RowObject temp-table; application of new naming standards

Demo

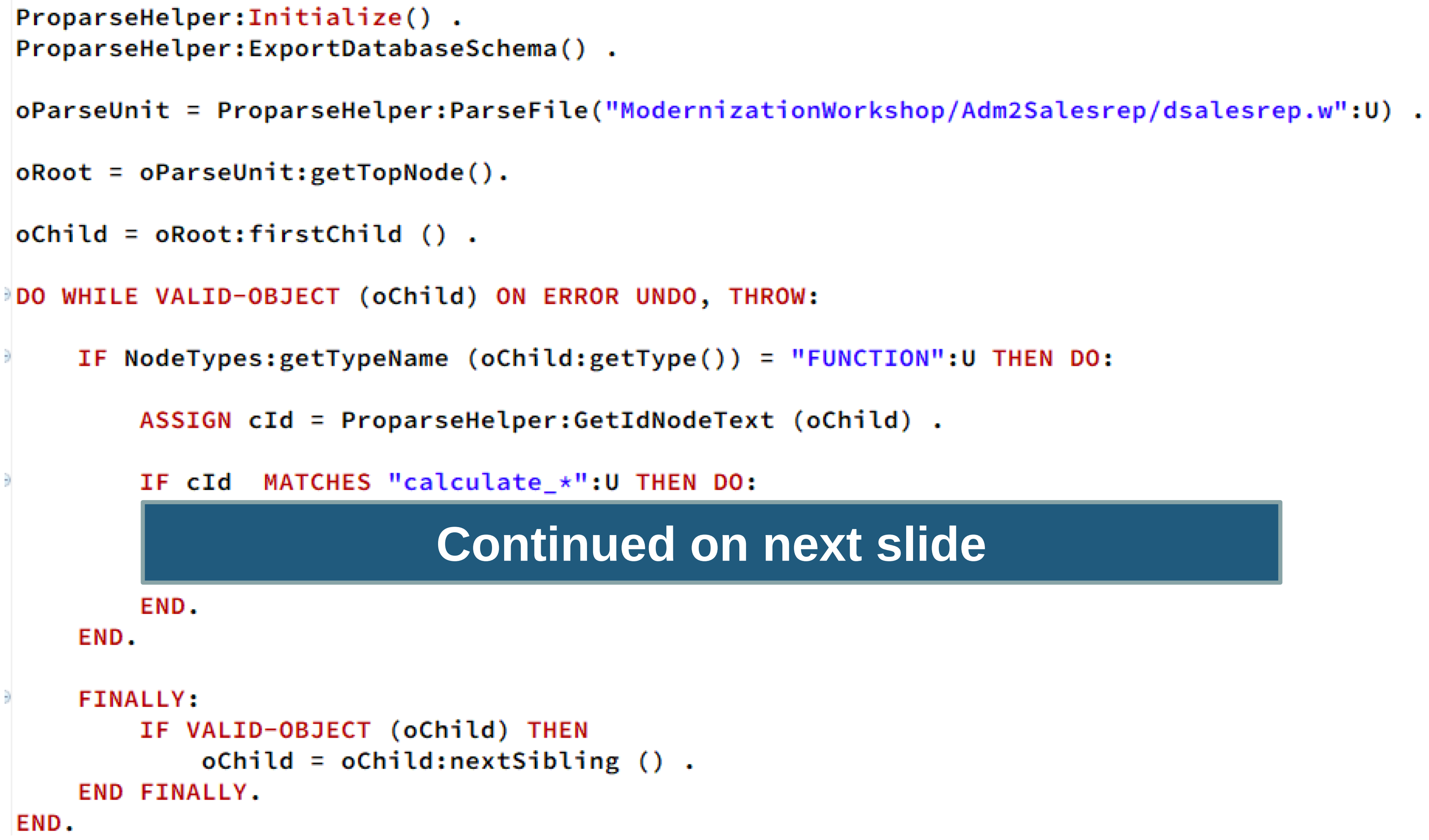
- Use Business Entity Designer plugin to convert SDO into Business Entity

Source Code migration

- Migration of arbitrary source-code influenced by existing coding style
- Migration of SDO source code requires
  - Location of relevant source code
  - Conversion of procedures/functions to methods
  - Modify procedural invocation of sub-routines to class based invocation
  - Change access to RowObject fields to new temp-table name
  - …

Proparse based source-code migration

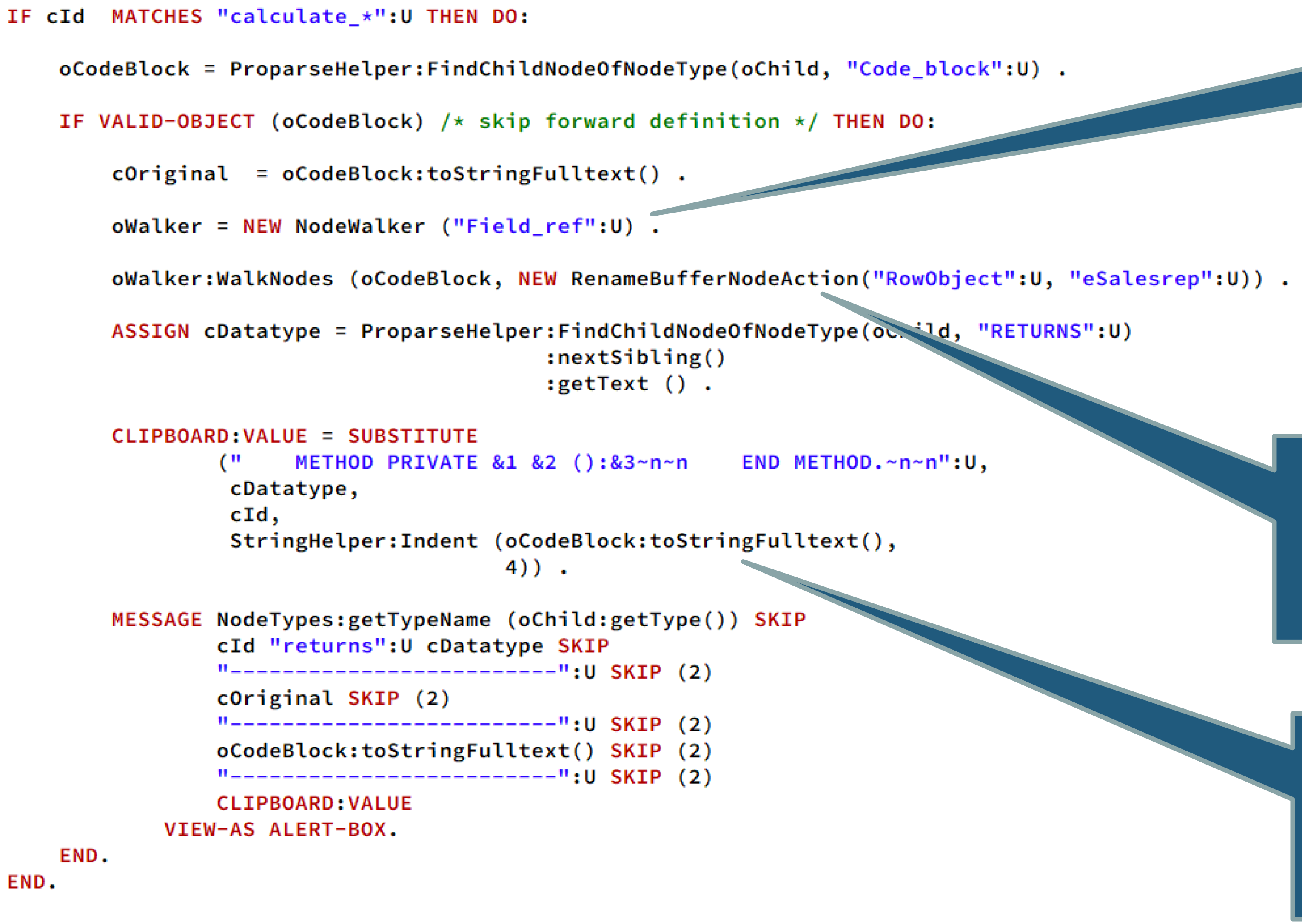
- Extension to Proparse
  - ABL based API’s to locate relevant code
  - enabling Proparse for in-memory manipulation of source code
- Alternative is to use Proparse for understanding of legacy code and simple OUTPUT TO or LONGCHAR operations to build new source code
- XFEF, COMPILE listing sometimes used as well. But majority of input is present in Proparse



```
ProparseHelper:Initialize() .
ProparseHelper:ExportDatabaseSchema() .

oParseUnit = ProparseHelper:ParseFile("ModernizationWorkshop/Adm2Salesrep/dsalesrep.w":U) .

oRoot = oParseUnit:getTopNode().

oChild = oRoot:firstChild () .

DO WHILE VALID-OBJECT (oChild) ON ERROR UNDO, THROW:

    IF NodeTypes:getTypeName (oChild:getType()) = "FUNCTION":U THEN DO:

        ASSIGN cId = ProparseHelper:GetIdNodeText (oChild) .

        IF cId  MATCHES "calculate_*":U THEN DO:
```

Continued on next slide

```
        END.
    END.

    FINALLY:
        IF VALID-OBJECT (oChild) THEN
            oChild = oChild:nextSibling () .
    END FINALLY.
END.
```

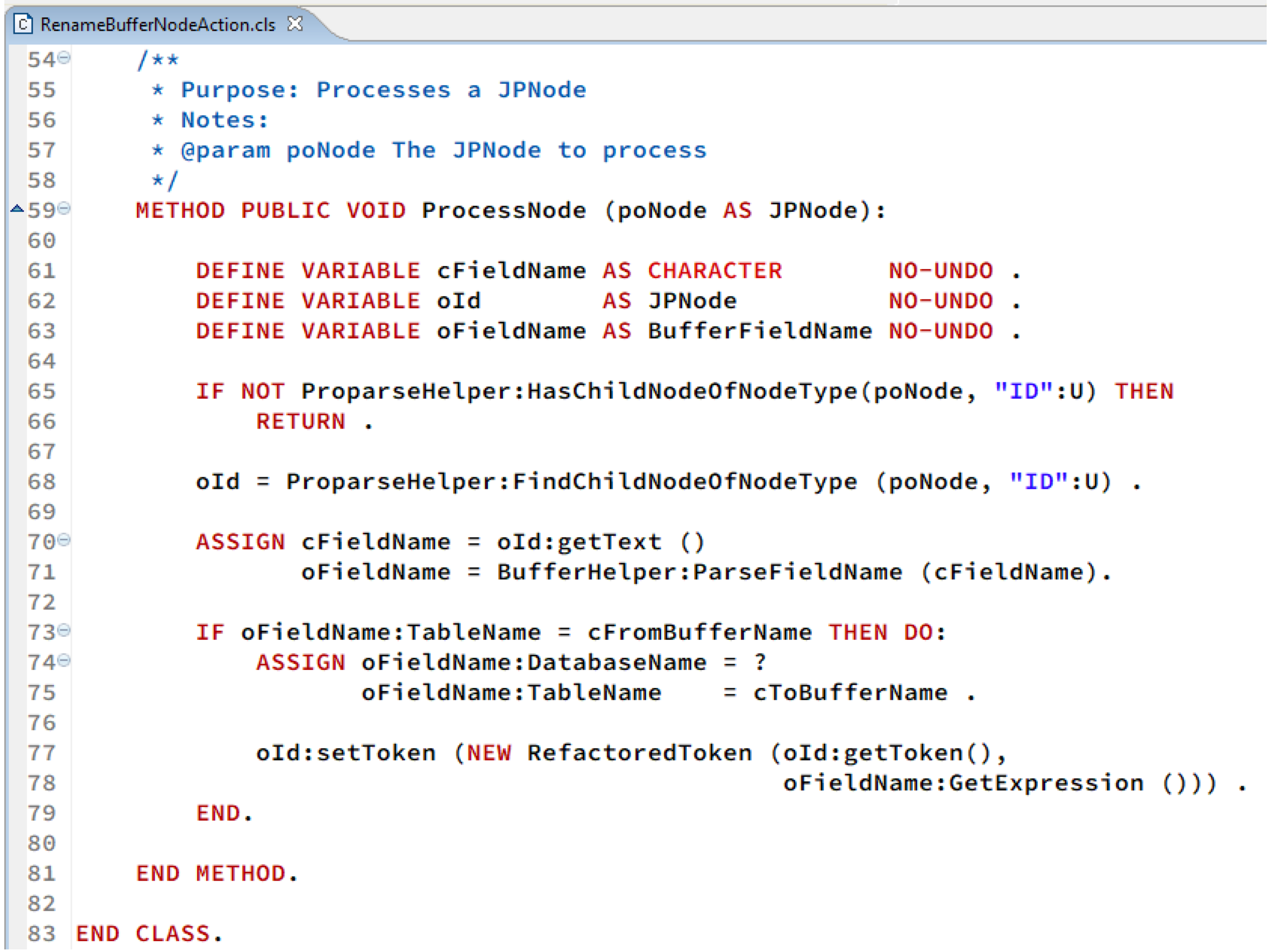


```
IF cId  MATCHES "calculate_*":U THEN DO:

    oCodeBlock = ProparseHelper:FindChildNodeOfNodeType(oChild, "Code_block":U) .

    IF VALID-OBJECT (oCodeBlock) /* skip forward definition */ THEN DO:

        cOriginal  = oCodeBlock:toStringFulltext() .

        oWalker = NEW NodeWalker ("Field_ref":U) .

        oWalker:WalkNodes (oCodeBlock, NEW RenameBufferNodeAction("RowObject":U, "eSalesrep":U)) .

        ASSIGN cDatatype = ProparseHelper:FindChildNodeOfNodeType(oChild, "RETURNS":U)
                                         :nextSibling()
                                         :getText () .

        CLIPBOARD:VALUE = SUBSTITUTE
                ("    METHOD PRIVATE &1 &2 ():&3~n~n    END METHOD.~n~n":U,
                 cDatatype,
                 cId,
                 StringHelper:Indent (oCodeBlock:toStringFulltext(),
                                      4)) .

        MESSAGE NodeTypes:getTypeName (oChild:getType()) SKIP
                cId "returns":U cDatatype SKIP
                "-------------------------":U SKIP (2)
                cOriginal SKIP (2)
                "-------------------------":U SKIP (2)
                oCodeBlock:toStringFulltext() SKIP (2)
                "-------------------------":U SKIP (2)
                CLIPBOARD:VALUE
            VIEW-AS ALERT-BOX.
    END.
END.
```

Recursively processes JPNodes

Injected into NodeWalker, rewrites RowObject references in AST

Returns modified function source code

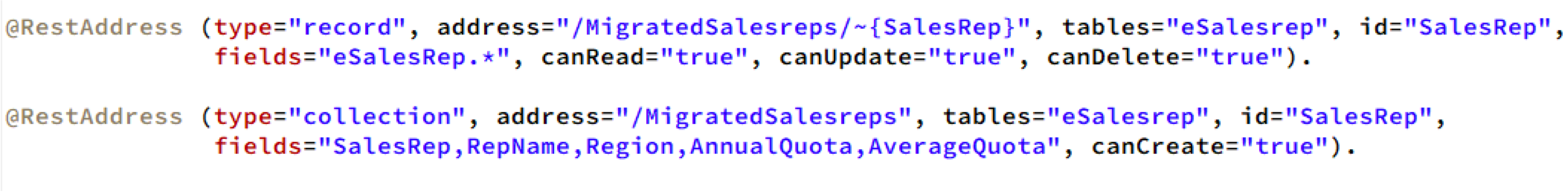


```
    /**
     * Purpose: Processes a JPNode
     * Notes:
     * @param poNode The JPNode to process
     */
    METHOD PUBLIC VOID ProcessNode (poNode AS JPNode):

        DEFINE VARIABLE cFieldName AS CHARACTER       NO-UNDO .
        DEFINE VARIABLE oId        AS JPNode          NO-UNDO .
        DEFINE VARIABLE oFieldName AS BufferFieldName NO-UNDO .

        IF NOT ProparseHelper:HasChildNodeOfNodeType(poNode, "ID":U) THEN
            RETURN .

        oId = ProparseHelper:FindChildNodeOfNodeType (poNode, "ID":U) .

        ASSIGN cFieldName = oId:getText ()
               oFieldName = BufferHelper:ParseFieldName (cFieldName).

        IF oFieldName:TableName = cFromBufferName THEN DO:
            ASSIGN oFieldName:DatabaseName = ?
                   oFieldName:TableName     = cToBufferName .

            oId:setToken (NEW RefactoredToken (oId:getToken(),
                                               oFieldName:GetExpression ())) .
        END.

    END METHOD.

END CLASS.
```

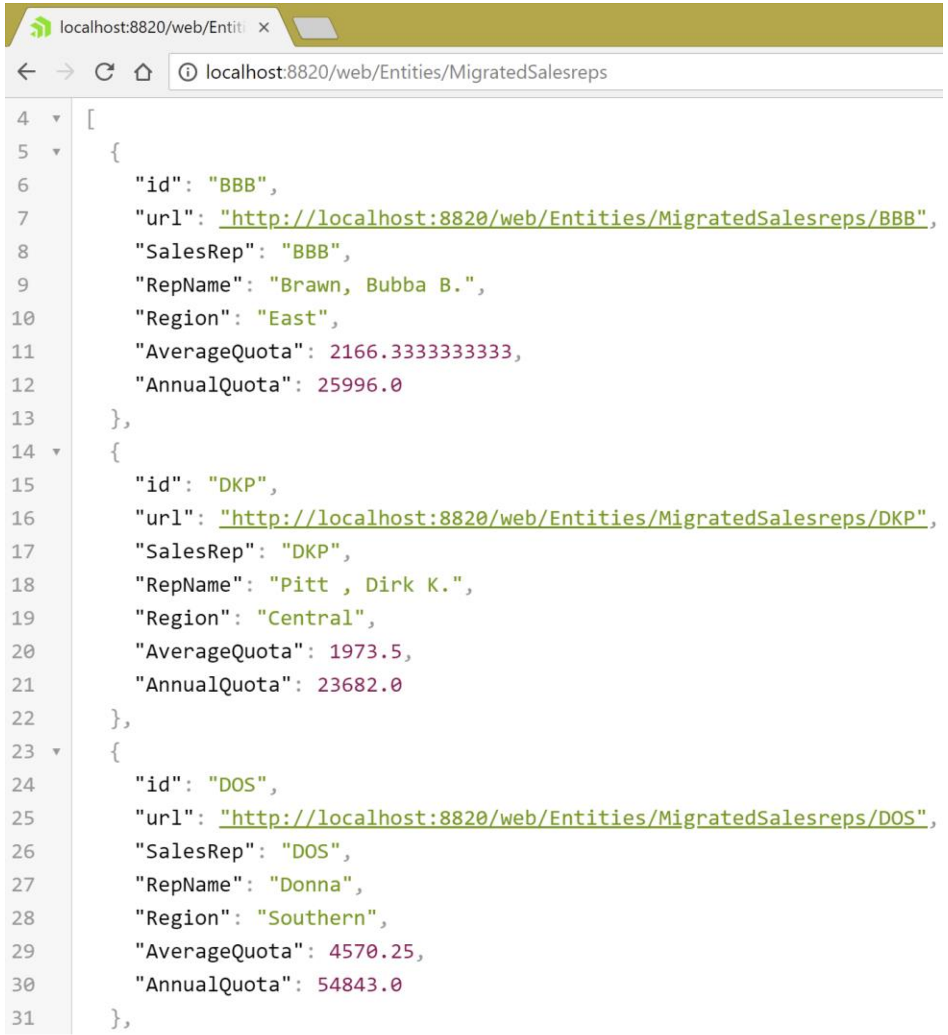
Demo

- Migration Routines for
  - Calculated Field source code
  - Validation Procedures
  - Test Business Entity / Calculated Fields in Business Entity Tester
  - Test Update and Validation using source code
  - Define RESTful Endpoint for the Business Entity

Define RESTful endpoints using Annotations



```
@RestAddress (type="record", address="/MigratedSalesreps/~{SalesRep}", tables="eSalesrep", id="SalesRep",
              fields="eSalesRep.*", canRead="true", canUpdate="true", canDelete="true").

@RestAddress (type="collection", address="/MigratedSalesreps", tables="eSalesrep", id="SalesRep",
              fields="SalesRep,RepName,Region,AnnualQuota,AverageQuota", canCreate="true").
```



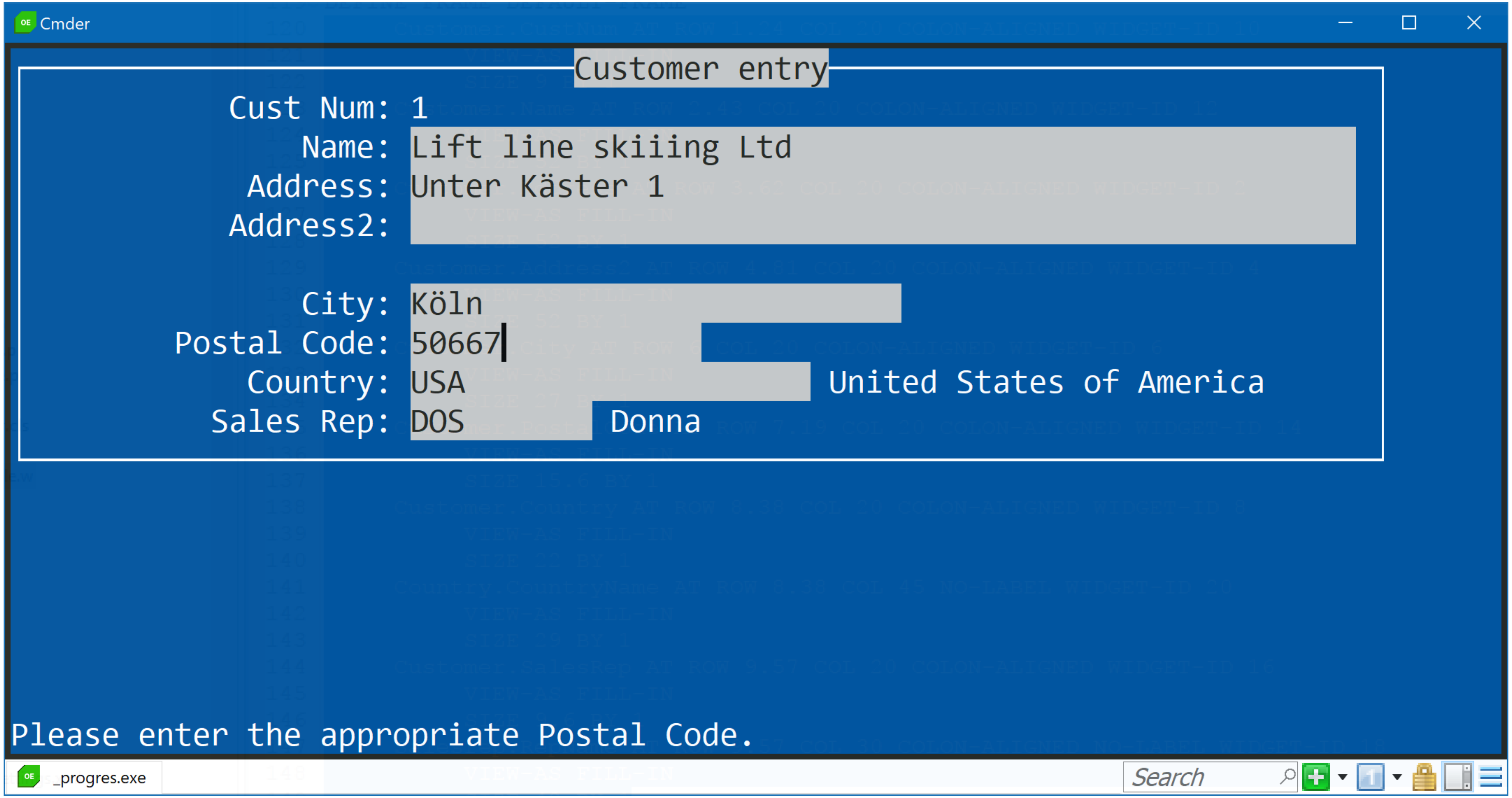
localhost:8820/web/Entiti ×

localhost:8820/web/Entities/MigratedSalesreps

```
[
  {
    "id": "BBB",
    "url": "http://localhost:8820/web/Entities/MigratedSalesreps/BBB",
    "SalesRep": "BBB",
    "RepName": "Brawn, Bubba B.",
    "Region": "East",
    "AverageQuota": 2166.3333333333,
    "AnnualQuota": 25996.0
  },
  {
    "id": "DKP",
    "url": "http://localhost:8820/web/Entities/MigratedSalesreps/DKP",
    "SalesRep": "DKP",
    "RepName": "Pitt , Dirk K.",
    "Region": "Central",
    "AverageQuota": 1973.5,
    "AnnualQuota": 23682.0
  },
  {
    "id": "DOS",
    "url": "http://localhost:8820/web/Entities/MigratedSalesreps/DOS",
    "SalesRep": "DOS",
    "RepName": "Donna",
    "Region": "Southern",
    "AverageQuota": 4570.25,
    "AnnualQuota": 54843.0
  },
```

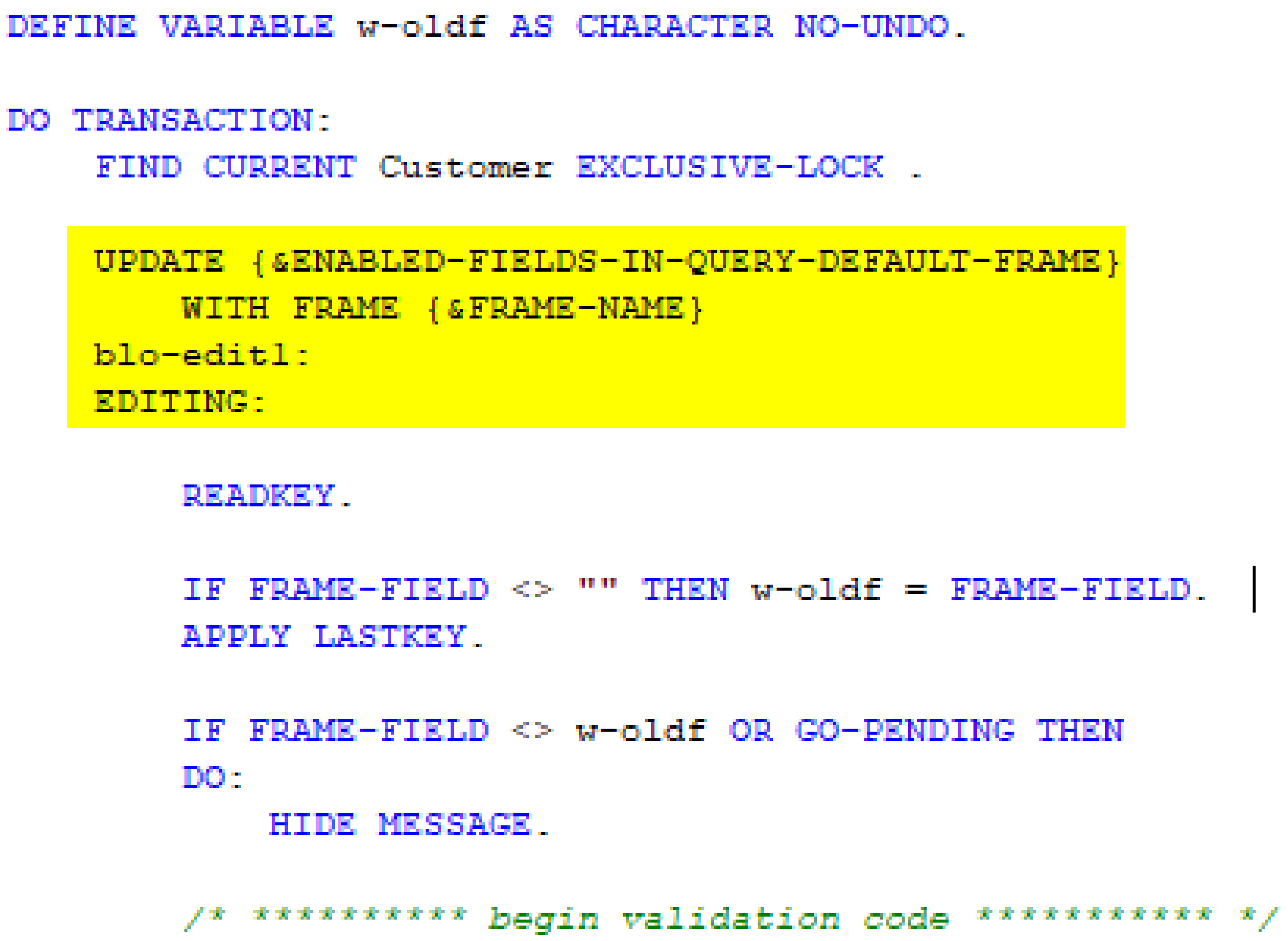
Agenda

- Introduction / Application Modernization
- Modern OpenEdge Application Architecture
- ADM2 SDO migration
- **TTY Upgrade editing migration**
- ABL GUI migration
- OSIV3G Modernization example



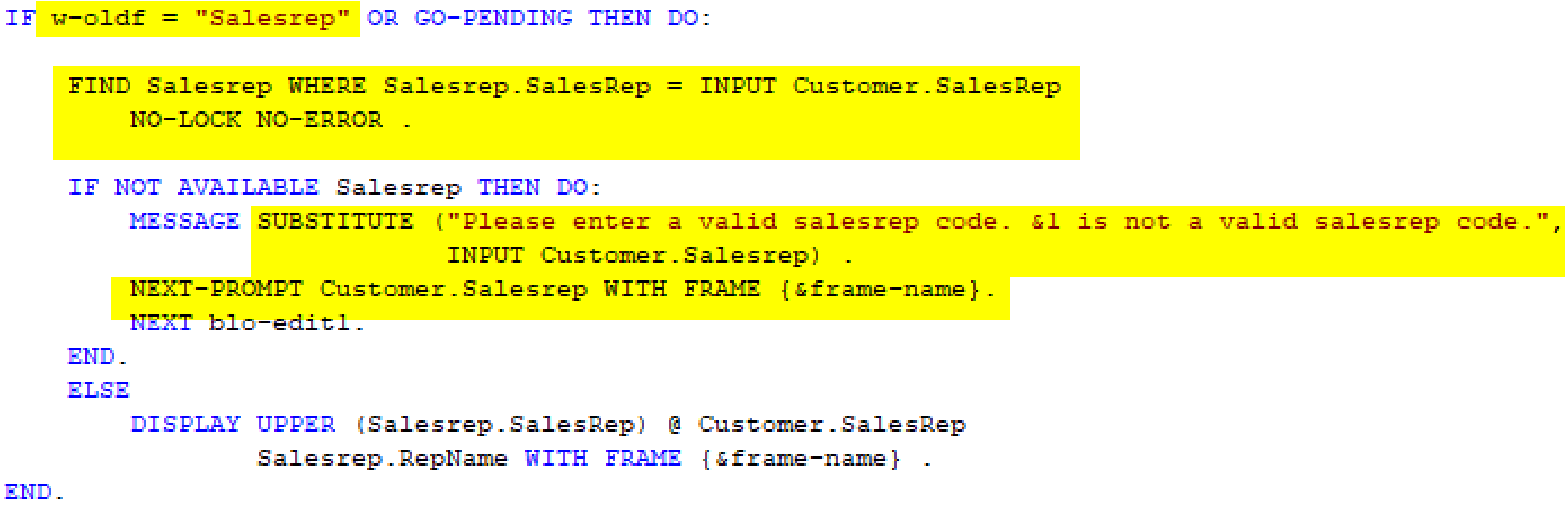
```
                      Customer entry
     Cust Num: 1
         Name: Lift line skiiing Ltd
      Address: Unter Käster 1
     Address2:

         City: Köln
  Postal Code: 50667
      Country: USA                      United States of America
    Sales Rep: DOS          Donna

Please enter the appropriate Postal Code.
```

UPDATE EDITING Blocks



```
DEFINE VARIABLE w-oldf AS CHARACTER NO-UNDO.

DO TRANSACTION:
    FIND CURRENT Customer EXCLUSIVE-LOCK .

    UPDATE {&ENABLED-FIELDS-IN-QUERY-DEFAULT-FRAME}
        WITH FRAME {&FRAME-NAME}
    blo-editl:
    EDITING:

        READKEY.

        IF FRAME-FIELD <> "" THEN w-oldf = FRAME-FIELD.
        APPLY LASTKEY.

        IF FRAME-FIELD <> w-oldf OR GO-PENDING THEN
        DO:
            HIDE MESSAGE.

        /* *********** begin validation code ************ */
```

Single field validation within EDITING Block



```
IF w-oldf = "Salesrep" OR GO-PENDING THEN DO:

    FIND Salesrep WHERE Salesrep.SalesRep = INPUT Customer.SalesRep
        NO-LOCK NO-ERROR .

    IF NOT AVAILABLE Salesrep THEN DO:
        MESSAGE SUBSTITUTE ("Please enter a valid salesrep code. &1 is not a valid salesrep code.",
                            INPUT Customer.Salesrep) .
        NEXT-PROMPT Customer.Salesrep WITH FRAME {&frame-name}.
        NEXT blo-edit1.
    END.
    ELSE
        DISPLAY UPPER (Salesrep.SalesRep) @ Customer.SalesRep
                Salesrep.RepName WITH FRAME {&frame-name} .
END.
```

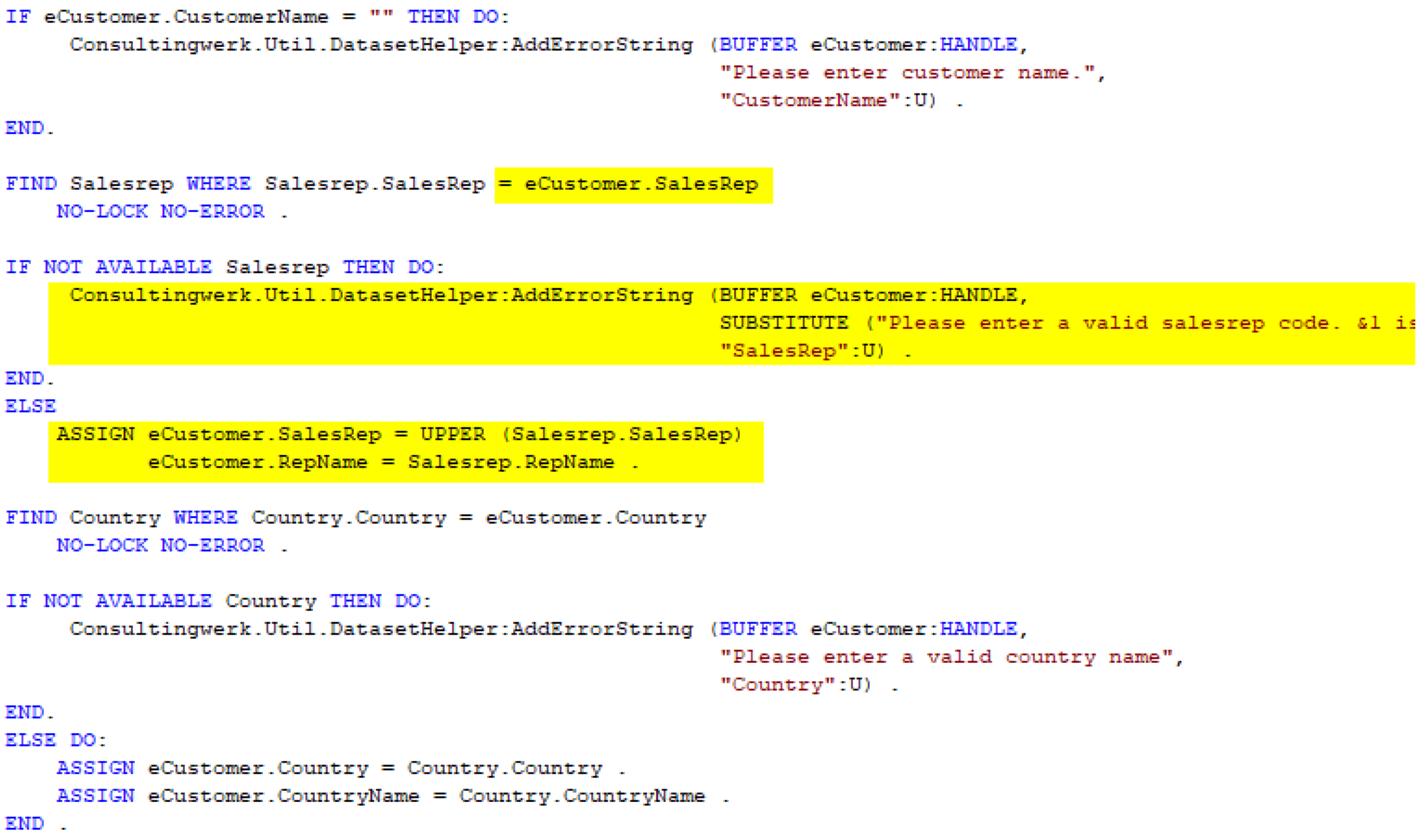
UPDATE EDITING Blocks

- Commonly used in TTY and early GUI applications
- Full of validation logic / Lookup functionality (locating foreign key descriptions)
- Tied to UI through “INPUT <fieldname>” references
- MESSAGE Statement used for error messages
- NEXT-PROMPT provides field that should receive input after error
- Record locked during duration of the UPDATE Statement

UPDATE EDITING Blocks

- Iterated for every keystroke or GO-PENDING
- When invoked on GO-PENDING, it’s similar to a commit to a Business Entity
  - Validating all fields at once
  - Processing update when no validation error occurred
  - Returning validation error to user (with instruction of next field)
- Code flow in EDITING Block very similar to typical Business Entity validation

Business Entity Validation based on UPD EDITING



```
IF eCustomer.CustomerName = "" THEN DO:
     Consultingwerk.Util.DatasetHelper:AddErrorString (BUFFER eCustomer:HANDLE,
                                                       "Please enter customer name.",
                                                       "CustomerName":U) .
END.

FIND Salesrep WHERE Salesrep.SalesRep = eCustomer.SalesRep
    NO-LOCK NO-ERROR .

IF NOT AVAILABLE Salesrep THEN DO:
     Consultingwerk.Util.DatasetHelper:AddErrorString (BUFFER eCustomer:HANDLE,
                                                       SUBSTITUTE ("Please enter a valid salesrep code. &1 is
                                                       "SalesRep":U) .
END.
ELSE
    ASSIGN eCustomer.SalesRep = UPPER (Salesrep.SalesRep)
           eCustomer.RepName = Salesrep.RepName .

FIND Country WHERE Country.Country = eCustomer.Country
    NO-LOCK NO-ERROR .

IF NOT AVAILABLE Country THEN DO:
     Consultingwerk.Util.DatasetHelper:AddErrorString (BUFFER eCustomer:HANDLE,
                                                       "Please enter a valid country name",
                                                       "Country":U) .
END.
ELSE DO:
    ASSIGN eCustomer.Country = Country.Country .
    ASSIGN eCustomer.CountryName = Country.CountryName .
END .
```

Business Entity Validation based on UPD EDITING

- IF w-oldf OR GO-ENDING not required; Business Entity typically validates all fields at once
  - Removing at least one level of blocks in the code
- *“INPUT <fieldname>”* replaced with temp-table field reference
- *DISPLAY* statements replaces with update of temp-table field
- *MESSAGE/NEXT-PROMPT* statements replaced with API call to return validation message to the consumer of the Business Entity and control target field

Demo

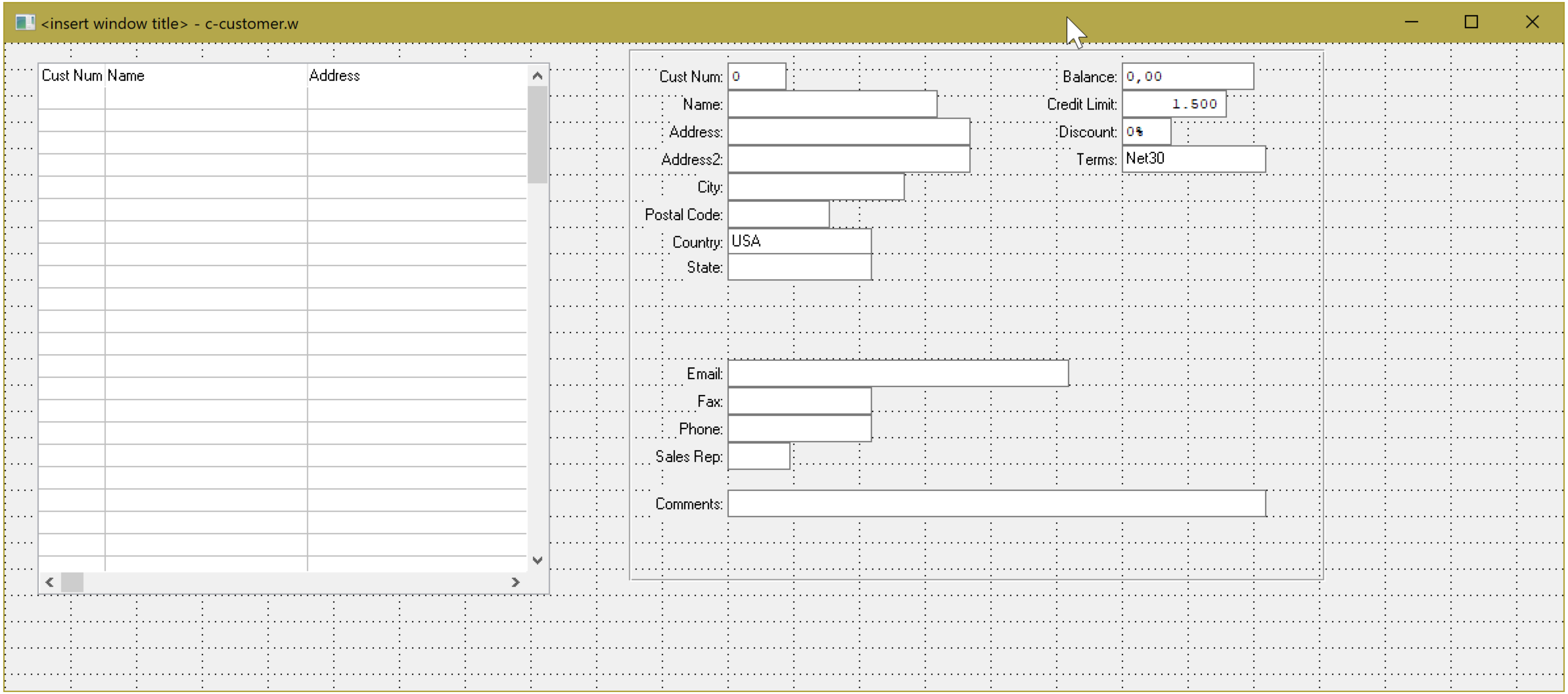
- Proparse based migration of UPDATE EDITING Blocks into Business Entity Validation block

Agenda

- Introduction / Application Modernization
- Modern OpenEdge Application Architecture
- ADM2 SDO migration
- TTY Upgrade editing migration
- **ABL GUI migration**
- OSIV3G Modernization example

ABL GUI Migration

- Existing GUI (or TTY) screen layout may serve as a starting point for new UI's
  - Highly dependent on UX of new application
  - Highly dependent on "quality" of layout of new application



Cust Num: 0

Name:

Address:

Address2:

City:

Postal Code:

Country: USA

State:

Email:

Fax:

Phone:

Sales Rep:

Comments:

Balance: 0,00

Credit Limit: 1.500

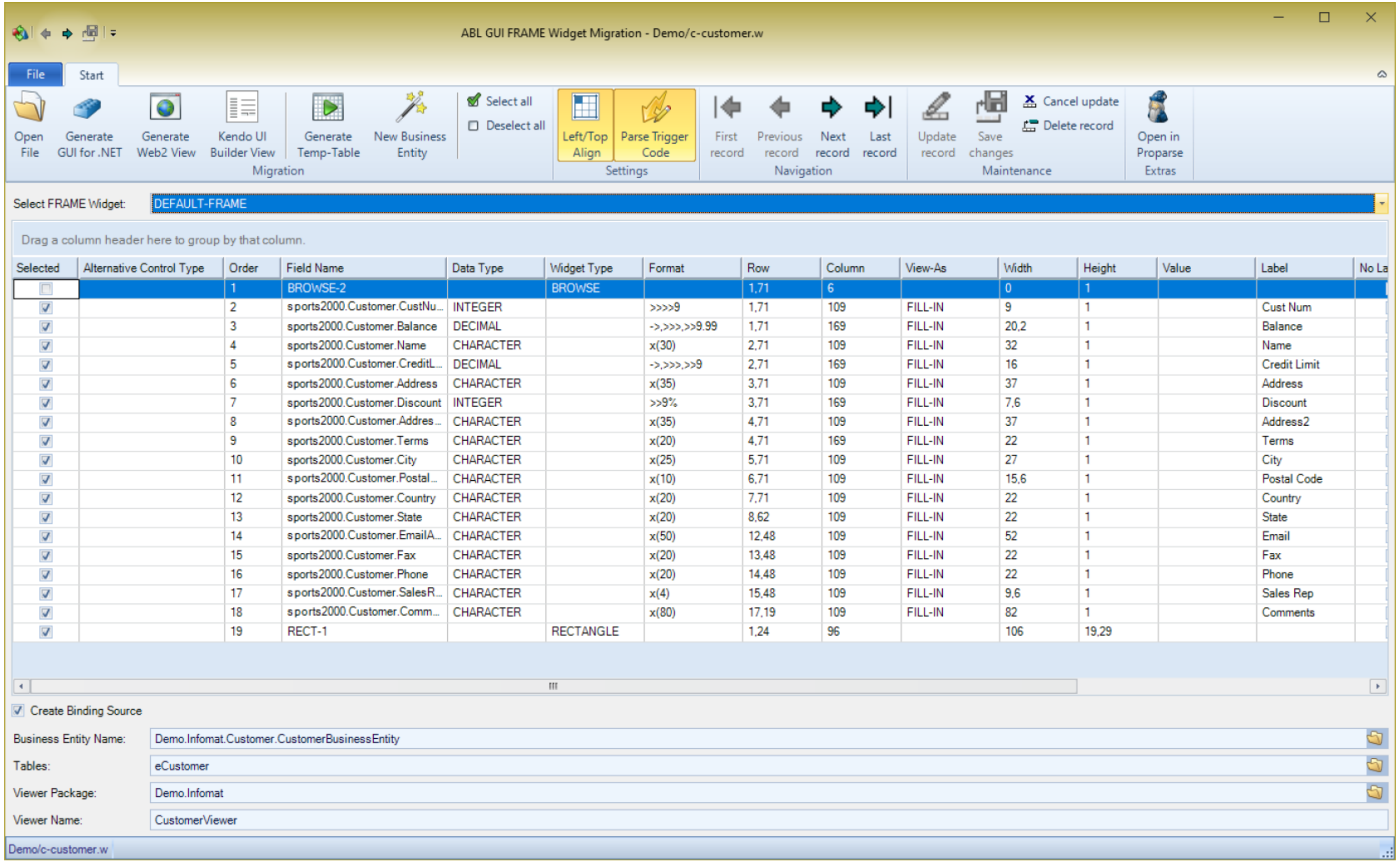
Discount: 0%

Terms: Net30

Screen layout migration

- Screen layout from static code can be refactored based on Proparse
  - FRAME definitions sometimes tricky to understand
  - Multiple FRAME Statements for a single FRAME
  - VIEW-AS phrase from Data Dictionary
  - Default properties of widgets
- Walking the widget tree typically simpler – however this requires changes to application runtime and is not trivial when building general purpose tools

Abstract view on screen layout



Select FRAME Widget: DEFAULT-FRAME

Selected	Alternative Control Type	Order	Field Name	Data Type	Widget Type	Format	Row	Column	View-As	Width	Height	Value	Label	No La
□		1	BROWSE-2		BROWSE		1,71	6		0	1			
☑		2	sports2000.Customer.CustNu...	INTEGER		>>>>9	1,71	109	FILL-IN	9	1		Cust Num	
☑		3	sports2000.Customer.Balance	DECIMAL		->,>>>,>>9.99	1,71	169	FILL-IN	20,2	1		Balance	
☑		4	sports2000.Customer.Name	CHARACTER		x(30)	2,71	109	FILL-IN	32	1		Name	
☑		5	sports2000.Customer.CreditL...	DECIMAL		->,>>>,>>9	2,71	169	FILL-IN	16	1		Credit Limit	
☑		6	sports2000.Customer.Address	CHARACTER		x(35)	3,71	109	FILL-IN	37	1		Address	
☑		7	sports2000.Customer.Discount	INTEGER		>>9%	3,71	169	FILL-IN	7,6	1		Discount	
☑		8	sports2000.Customer.Addres...	CHARACTER		x(35)	4,71	109	FILL-IN	37	1		Address2	
☑		9	sports2000.Customer.Terms	CHARACTER		x(20)	4,71	169	FILL-IN	22	1		Terms	
☑		10	sports2000.Customer.City	CHARACTER		x(25)	5,71	109	FILL-IN	27	1		City	
☑		11	sports2000.Customer.Postal...	CHARACTER		x(10)	6,71	109	FILL-IN	15,6	1		Postal Code	
☑		12	sports2000.Customer.Country	CHARACTER		x(20)	7,71	109	FILL-IN	22	1		Country	
☑		13	sports2000.Customer.State	CHARACTER		x(20)	8,62	109	FILL-IN	22	1		State	
☑		14	sports2000.Customer.EmailA...	CHARACTER		x(50)	12,48	109	FILL-IN	52	1		Email	
☑		15	sports2000.Customer.Fax	CHARACTER		x(20)	13,48	109	FILL-IN	22	1		Fax	
☑		16	sports2000.Customer.Phone	CHARACTER		x(20)	14,48	109	FILL-IN	22	1		Phone	
☑		17	sports2000.Customer.SalesR...	CHARACTER		x(4)	15,48	109	FILL-IN	9,6	1		Sales Rep	
☑		18	sports2000.Customer.Comm...	CHARACTER		x(80)	17,19	109	FILL-IN	82	1		Comments	
☑		19	RECT-1		RECTANGLE		1,24	96		106	19,29			

☑ Create Binding Source

Business Entity Name: Demo.Infomat.Customer.CustomerBusinessEntity

Tables: eCustomer

Viewer Package: Demo.Infomat

Viewer Name: CustomerViewer

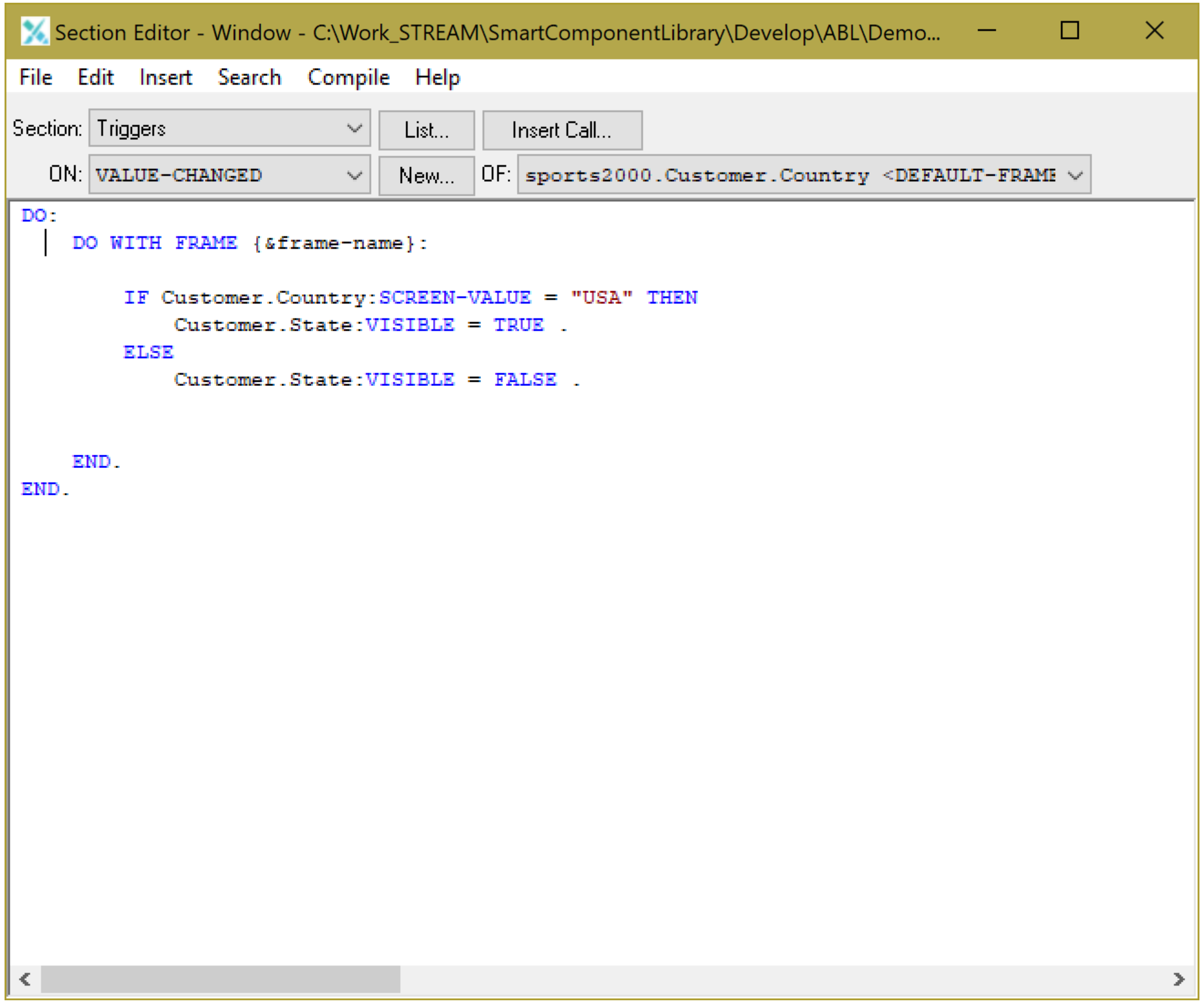
Demo/c-customer.w

Abstract view on screen layout

- Allows generation of various UI's
  - GUI for .NET
  - Angular
  - Kendo UI Builder
  - Meta-Data for UI repository database
  - …

GUI Trigger Code

- Typically used for validation or control of the UI
- Contains references using widget attributes (:SCREEN-VALUE or :SENSITIVE, etc.) or INPUT <fieldref>
- May contain business logic that should be moved to Business Entity (typically when accessing DB records), LEAVE Triggers typical prospect for validation


Section Editor - Window - C:\Work_STREAM\SmartComponentLibrary\Develop\ABL\Demo...
File Edit Insert Search Compile Help
List...
Insert Call...
New...

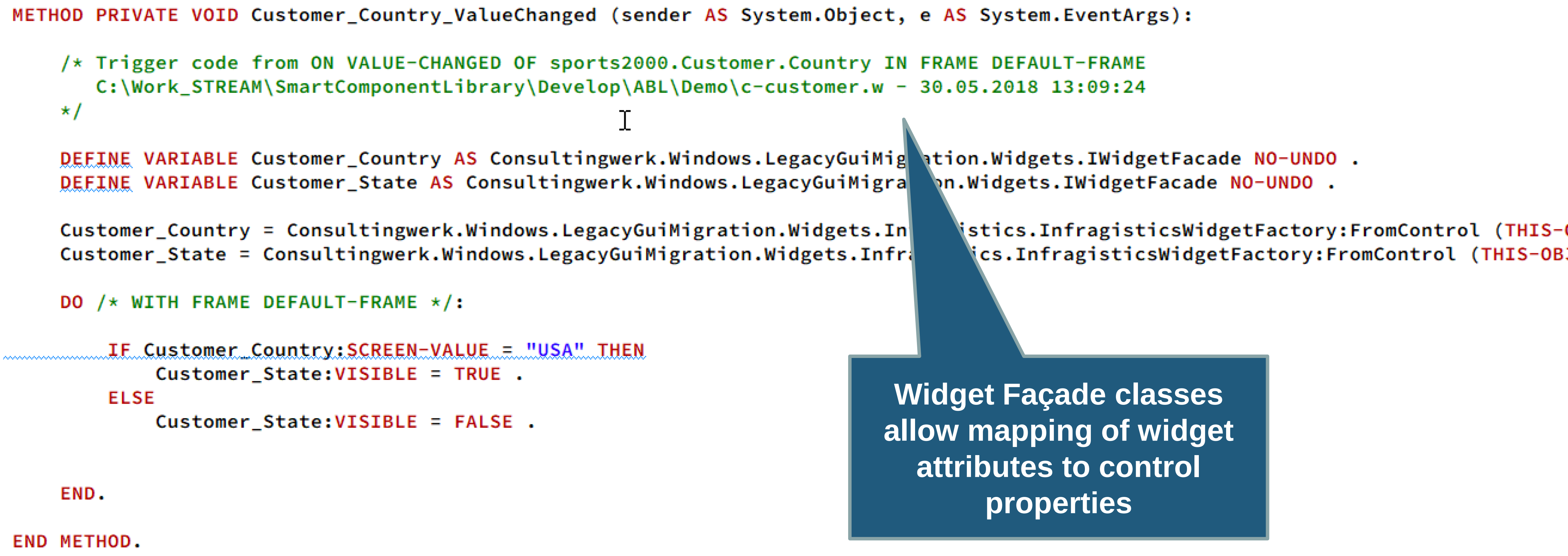
Section: Triggers

ON: VALUE-CHANGED

OF: sports2000.Customer.Country <DEFAULT-FRAME

```
DO:
    DO WITH FRAME {&frame-name}:

        IF Customer.Country:SCREEN-VALUE = "USA" THEN
            Customer.State:VISIBLE = TRUE .
        ELSE
            Customer.State:VISIBLE = FALSE .


    END.
END.
```

Migrated Trigger Code



```
METHOD PRIVATE VOID Customer_Country_ValueChanged (sender AS System.Object, e AS System.EventArgs):

    /* Trigger code from ON VALUE-CHANGED OF sports2000.Customer.Country IN FRAME DEFAULT-FRAME
       C:\Work_STREAM\SmartComponentLibrary\Develop\ABL\Demo\c-customer.w - 30.05.2018 13:09:24
    */

    DEFINE VARIABLE Customer_Country AS Consultingwerk.Windows.LegacyGuiMig ation.Widgets.IWidgetFacade NO-UNDO .
    DEFINE VARIABLE Customer_State AS Consultingwerk.Windows.LegacyGuiMigra n.Widgets.IWidgetFacade NO-UNDO .

    Customer_Country = Consultingwerk.Windows.LegacyGuiMigration.Widgets.In istics.InfragisticsWidgetFactory:FromControl (THIS-(
    Customer_State = Consultingwerk.Windows.LegacyGuiMigration.Widgets.Infra ics.InfragisticsWidgetFactory:FromControl (THIS-OB

    DO /* WITH FRAME DEFAULT-FRAME */:

        IF Customer_Country:SCREEN-VALUE = "USA" THEN
            Customer_State:VISIBLE = TRUE .
        ELSE
            Customer_State:VISIBLE = FALSE .


    END.

END METHOD.
```

Widget Façade classes allow mapping of widget attributes to control properties

Agenda

- Introduction / Application Modernization
- Modern OpenEdge Application Architecture
- ADM2 SDO migration
- TTY Upgrade editing migration
- ABL GUI migration
- **OSIV3G Modernization example**

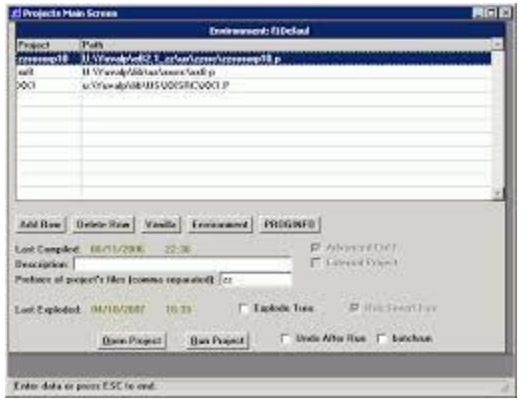
OSIV / OSC

- OSIV Service Center: Joint venture of 7 Swiss counties (cantons)
- Maintaining state insurance for occupational disabilities
- Approval of therapies
- Perform Disability and treatment Assessments
- Billing (by doctors, clinics, opticians, occupational disabilities, etc.)
- Document management
- 1300 users
- Very specific domain functionality
- Accepted by the user base, no real competition

Why “refactoring”

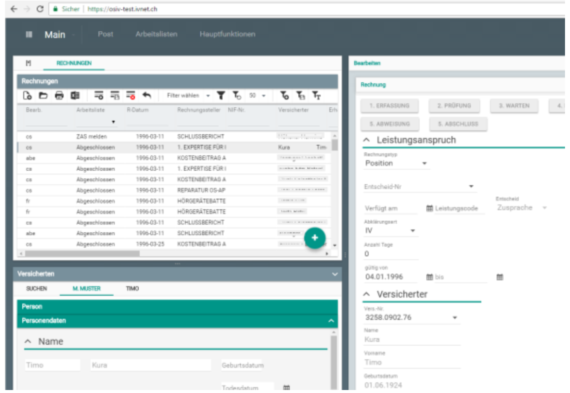
- Maintenance effort high
- Training of new users and developers hard
- Aged technology
- Resources / Motivation of developers / Agile methods

OSIV3G: Soft Migration



Current OSIV 5.x

Harvesting of existing Code

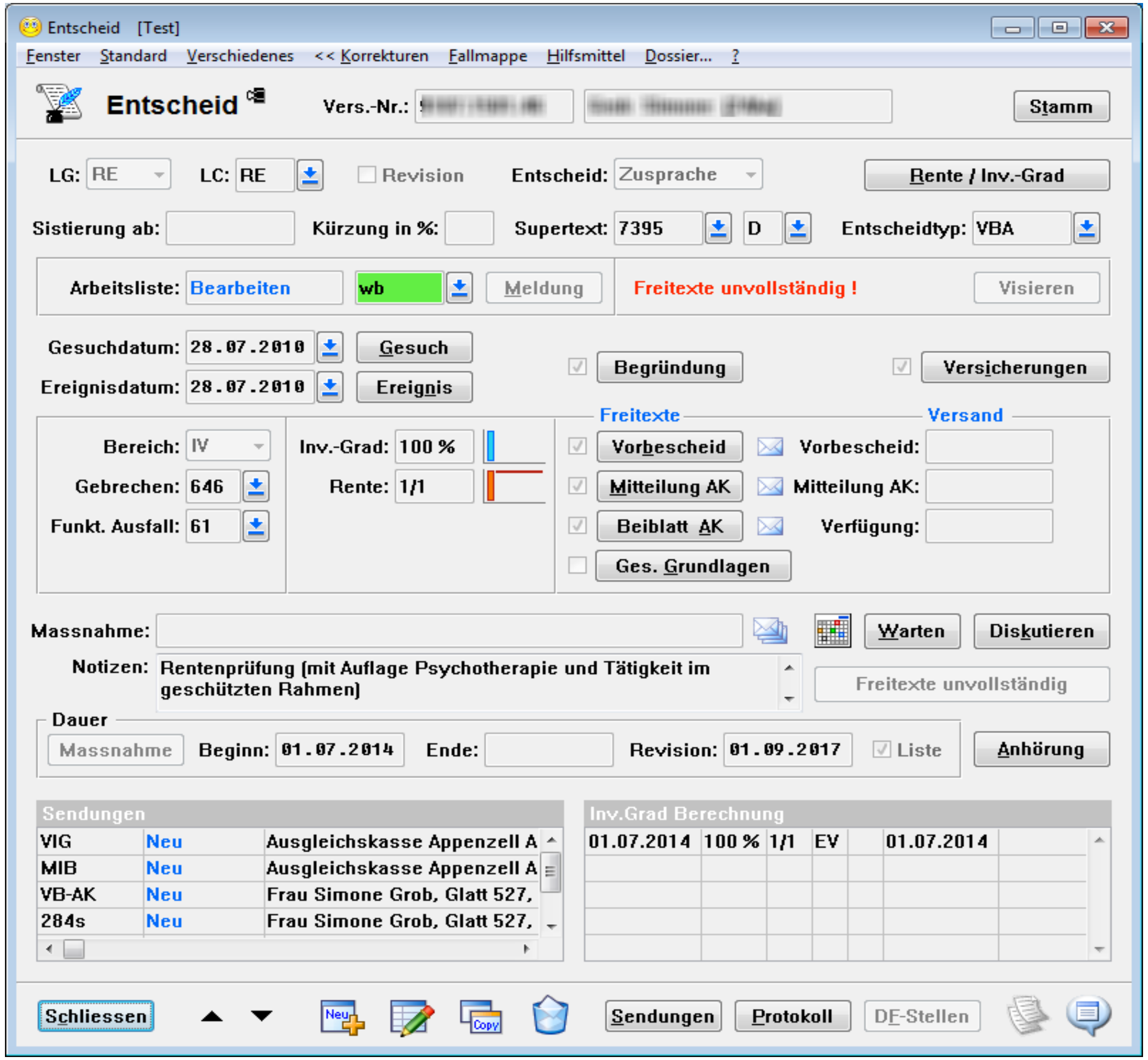


Migrated Application Modules

OSIV-DB

+ additional Fields and Tables e.g. GUID‘s

Framework DB‘s



Entscheid

Vers.-Nr.: [redacted] [redacted]

LG: RE LC: RE □ Revision Entscheid: Zusprache

Sistierung ab: Kürzung in %: Supertext: 7395 D Entscheidtyp: VBA

Arbeitsliste: Bearbeiten wb

Freitexte unvollständig !

Gesuchdatum: 28.07.2010

Ereignisdatum: 28.07.2010

Bereich: IV Inv.-Grad: 100 %

Gebrechen: 646 Rente: 1/1

Funkt. Ausfall: 61

Freitexte

☑ Begründung

☑ Vorbescheid

☑ Mitteilung AK

☑ Beiblatt AK

□ Ges. Grundlagen

☑ Versicherungen

Versand

Vorbescheid:

Mitteilung AK:

Verfügung:

Massnahme:

Notizen: Rentenprüfung (mit Auflage Psychotherapie und Tätigkeit im geschützten Rahmen)

Freitexte unvollständig

Dauer

Beginn: 01.07.2014 Ende: Revision: 01.09.2017 ☑ Liste

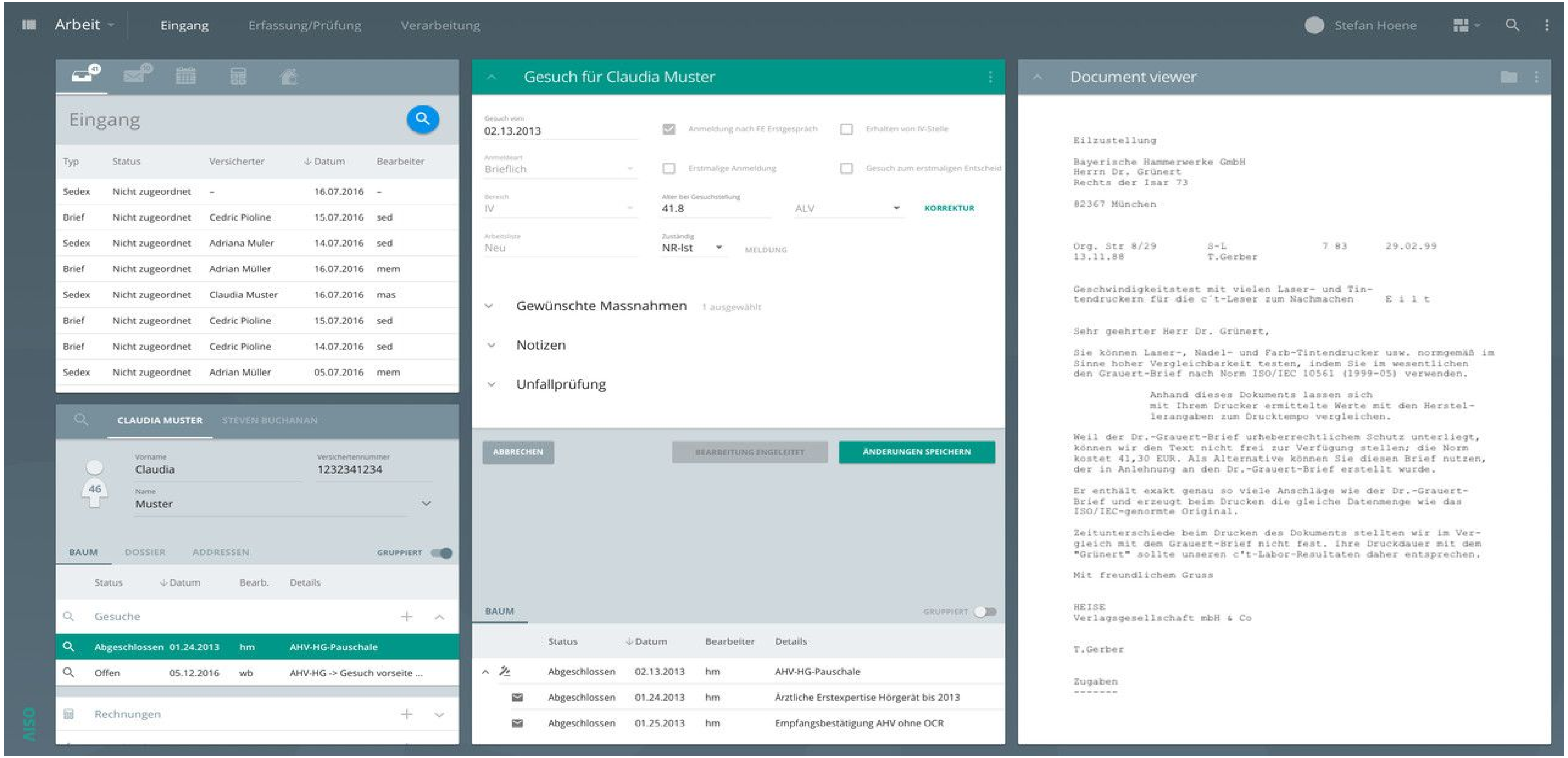
Sendungen

VIG	Neu	Ausgleichskasse Appenzell A
MIB	Neu	Ausgleichskasse Appenzell A
VB-AK	Neu	Frau Simone Grob, Glatt 527,
284s	Neu	Frau Simone Grob, Glatt 527,

Inv.Grad Berechnung

01.07.2014	100 %	1/1	EV	01.07.2014

Sneak Preview



Eingang

Typ	Status	Versicherter	↓ Datum	Bearbeiter
Sedex	Nicht zugeordnet	–	16.07.2016	–
Brief	Nicht zugeordnet	Cedric Pioline	15.07.2016	sed
Sedex	Nicht zugeordnet	Adriana Muler	14.07.2016	sed
Brief	Nicht zugeordnet	Adrian Müller	16.07.2016	mem
Sedex	Nicht zugeordnet	Claudia Muster	16.07.2016	mas
Brief	Nicht zugeordnet	Cedric Pioline	15.07.2016	sed
Brief	Nicht zugeordnet	Cedric Pioline	14.07.2016	sed
Sedex	Nicht zugeordnet	Adrian Müller	05.07.2016	mem

CLAUDIA MUSTER STEVEN BUCHANAN

46

Vorname: Claudia

Versichertennummer: 1232341234

Name: Muster

BAUM DOSSIER ADDRESSEN GRUPPIERT

Status	↓ Datum	Bearb.	Details
Gesuche			
Abgeschlossen	01.24.2013	hm	AHV-HG-Pauschale
Offen	05.12.2016	wb	AHV-HG -> Gesuch vorseite ...
Rechnungen			

Gesuch für Claudia Muster

Gesuch vom: 02.13.2013

☑ Anmeldung nach FE Erstgespräch

□ Erhalten von IV-Stelle

Anmeldeart: Brieflich

□ Erstmalige Anmeldung

□ Gesuch zum erstmaligen Entscheid

Bereich: IV

Alter bei Gesuchstellung: 41.8

ALV

KORREKTUR

Arbeitsliste: Neu

Zuständig: NR-Ist

MELDUNG

Gewünschte Massnahmen 1 ausgewählt

Notizen

Unfallprüfung

ABBRECHEN BEARBEITUNG ENGELEITET ÄNDERUNGEN SPEICHERN

BAUM GRUPPIERT

Status	↓ Datum	Bearbeiter	Details
Abgeschlossen	02.13.2013	hm	AHV-HG-Pauschale
Abgeschlossen	01.24.2013	hm	Ärztliche Erstexpertise Hörgerät bis 2013
Abgeschlossen	01.25.2013	hm	Empfangsbestätigung AHV ohne OCR

Document viewer

```
Eilzustellung

Bayerische Hammerwerke GmbH
Herrn Dr. Grünert
Rechts der Isar 73

82367 München



Org. Str 8/29          S-L                  7 83      29.02.99
13.11.88               T.Gerber


Geschwindigkeitstest mit vielen Laser- und Tin-
tendruckern für die c´t-Leser zum Nachmachen     E i l t


Sehr geehrter Herr Dr. Grünert,

Sie können Laser-, Nadel- und Farb-Tintendrucker usw. normgemäß im
Sinne hoher Vergleichbarkeit testen, indem Sie im wesentlichen
den Grauert-Brief nach Norm ISO/IEC 10561 (1999-05) verwenden.

            Anhand dieses Dokuments lassen sich
            mit Ihrem Drucker ermittelte Werte mit den Herstel-
            lerangaben zum Drucktempo vergleichen.

Weil der Dr.-Grauert-Brief urheberrechtlichem Schutz unterliegt,
können wir den Text nicht frei zur Verfügung stellen; die Norm
kostet 41,30 EUR. Als Alternative können Sie diesen Brief nutzen,
der in Anlehnung an den Dr.-Grauert-Brief erstellt wurde.

Er enthält exakt genau so viele Anschläge wie der Dr.-Grauert-
Brief und erzeugt beim Drucken die gleiche Datenmenge wie das
ISO/IEC-genormte Original.

Zeitunterschiede beim Drucken des Dokuments stellten wir im Ver-
gleich mit dem Grauert-Brief nicht fest. Ihre Druckdauer mit dem
"Grünert" sollte unseren c't-Labor-Resultaten daher entsprechen.

Mit freundlichem Gruss


HEISE
Verlagsgesellschaft mbH & Co


T.Gerber


Zugaben
-------
```

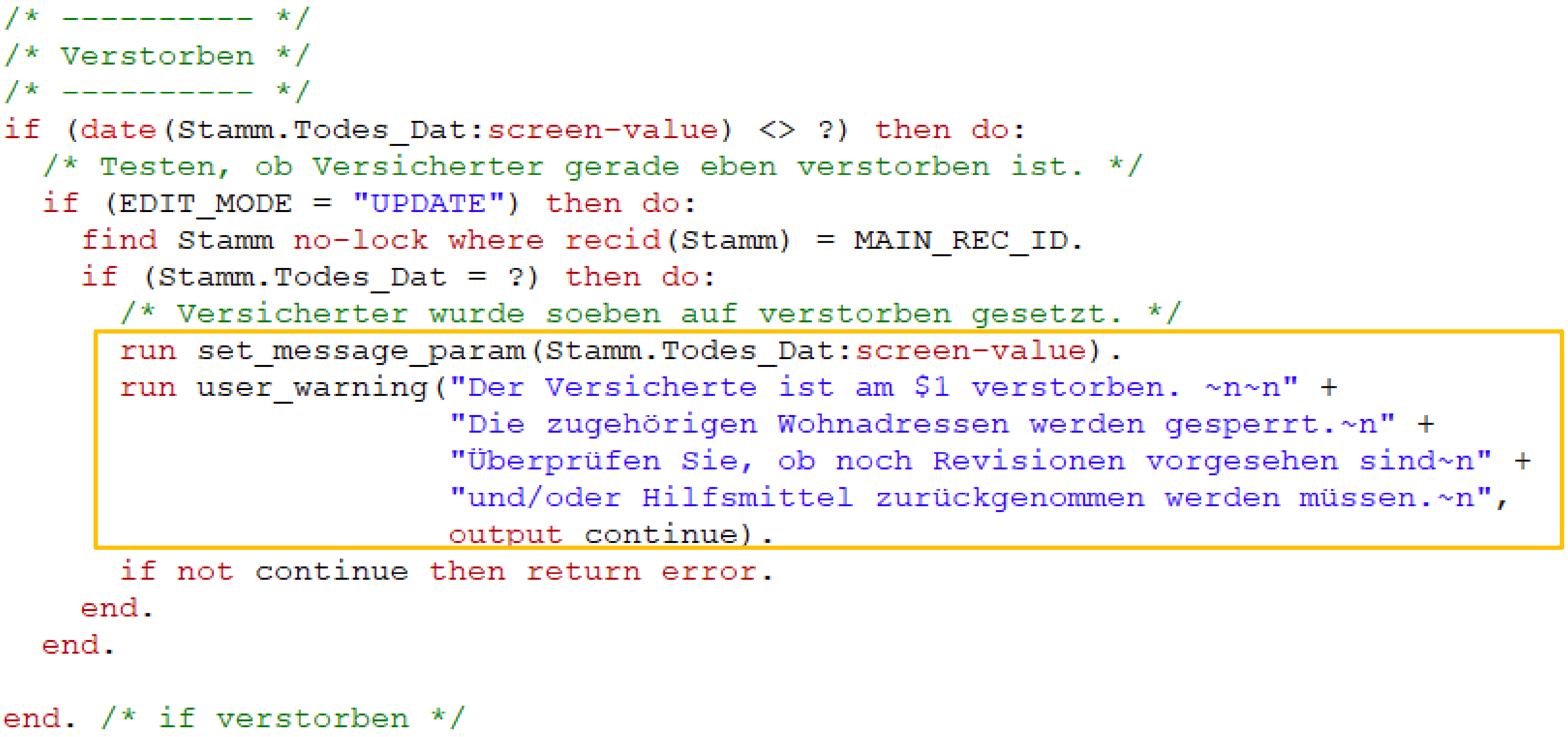
Example challenge: Interaction between Back and Frontend

- Existing OSIV Business Logic in large parts suitable as foundation for new OSIV3G (functional and structural), especially validation
- Validation may also provide color coding to represent field status etc.
- Validation may have to prompt the user
- Web applications typically:
  Request (from browser) – Response (from server)
- No Input-Blocking (not possible to wait for user input in Business Logic)

Sample: Yes/No PROMPT in validation

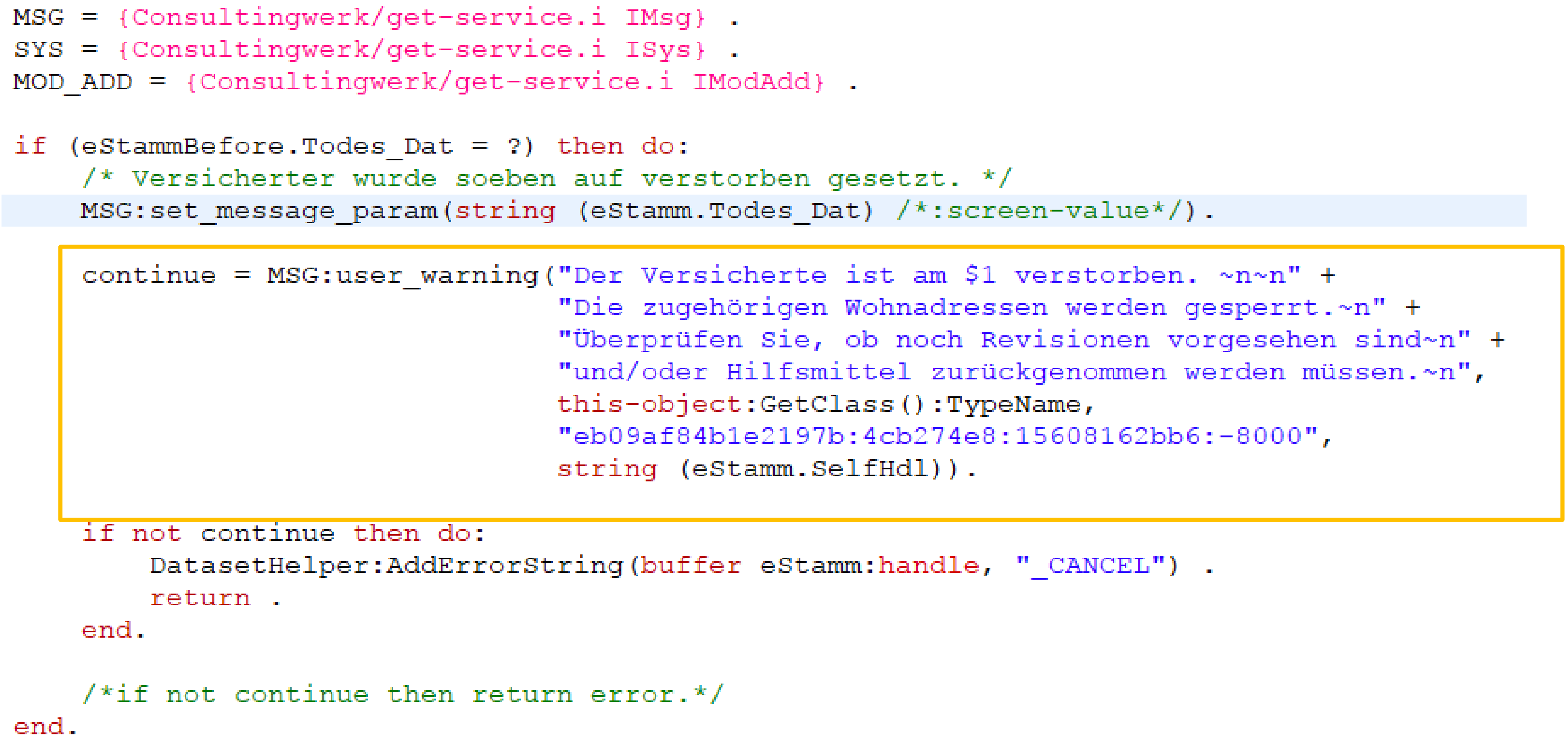
- Demand is to keep the validation flow in major parts „as is“
- Validation may encounter question requiring user input: “Are you sure?” etc.

Sample: Yes/No PROMPT in validation



```
/* ---------- */
/* Verstorben */
/* ---------- */
if (date(Stamm.Todes_Dat:screen-value) <> ?) then do:
  /* Testen, ob Versicherter gerade eben verstorben ist. */
  if (EDIT_MODE = "UPDATE") then do:
    find Stamm no-lock where recid(Stamm) = MAIN_REC_ID.
    if (Stamm.Todes_Dat = ?) then do:
      /* Versicherter wurde soeben auf verstorben gesetzt. */
      run set_message_param(Stamm.Todes_Dat:screen-value).
      run user_warning("Der Versicherte ist am $1 verstorben. ~n~n" +
                       "Die zugehörigen Wohnadressen werden gesperrt.~n" +
                       "Überprüfen Sie, ob noch Revisionen vorgesehen sind~n" +
                       "und/oder Hilfsmittel zurückgenommen werden müssen.~n",
                       output continue).
      if not continue then return error.
    end.
  end.

end. /* if verstorben */
```

Sample: Yes/No PROMPT in validation



```
MSG = {Consultingwerk/get-service.i IMsg} .
SYS = {Consultingwerk/get-service.i ISys} .
MOD_ADD = {Consultingwerk/get-service.i IModAdd} .

if (eStammBefore.Todes_Dat = ?) then do:
    /* Versicherter wurde soeben auf verstorben gesetzt. */
    MSG:set_message_param(string (eStamm.Todes_Dat) /*:screen-value*/).

    continue = MSG:user_warning("Der Versicherte ist am $1 verstorben. ~n~n" +
                                "Die zugehörigen Wohnadressen werden gesperrt.~n" +
                                "Überprüfen Sie, ob noch Revisionen vorgesehen sind~n" +
                                "und/oder Hilfsmittel zurückgenommen werden müssen.~n",
                                this-object:GetClass():TypeName,
                                "eb09af84b1e2197b:4cb274e8:15608162bb6:-8000",
                                string (eStamm.SelfHdl)).

    if not continue then do:
        DatasetHelper:AddErrorString(buffer eStamm:handle, "_CANCEL") .
        return .
    end.

    /*if not continue then return error.*/
end.
```

Consultingwerk
software architecture and development

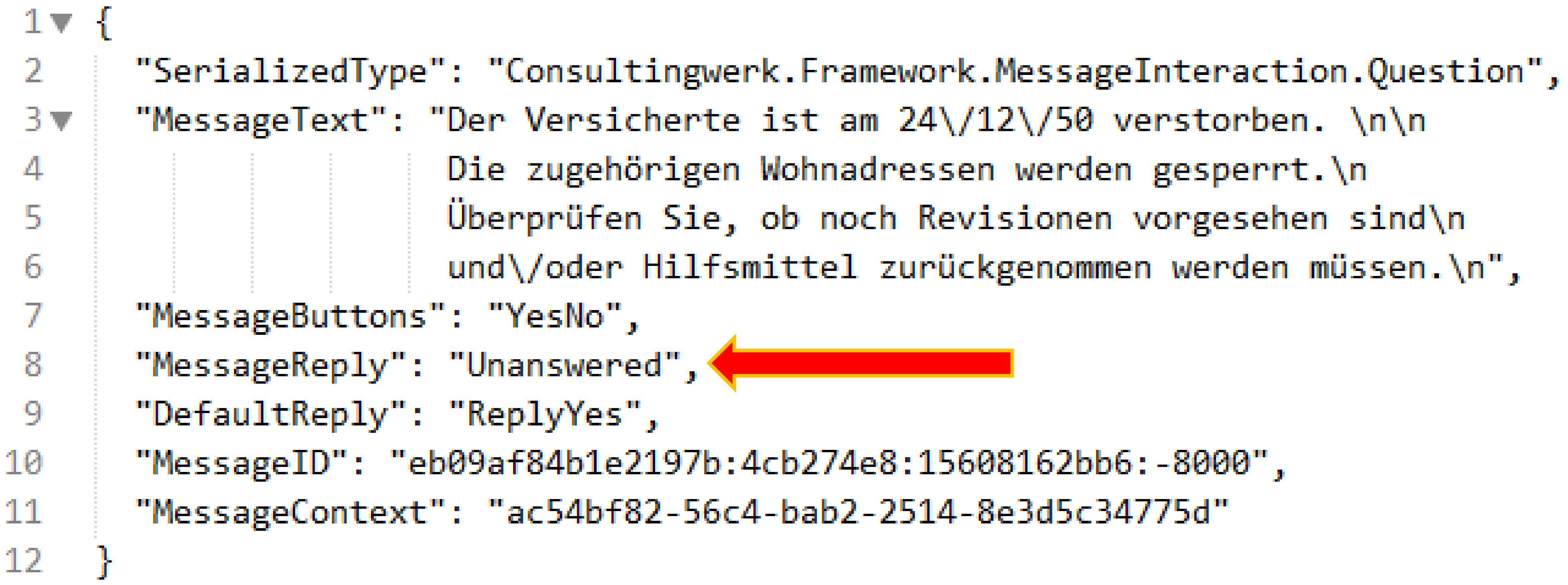
Migration using MessagePrompt API (SCL)

- Backend – API maintains list of questions (unanswered and answered)
- Same API Call may ask a new question of return an existing answer
- Supports multiple questions per routine: Questions are flagged with GUID identifying thise location in code
- Support for multiple iterations (Loops, FOR EACH, …): Each question is also flagged with a return PUK value

Migration using MessagePrompt API (SCL)

- Questions will be returned to UI in a standard temp-table field
- Current Update-Request will be cancelled (typically before the DB transaction is started)
- UI presents unanswered questions to the user and repeats the same update request
- Repeat this flow if additional questions are required

JSON Representation of the question



```
{
    "SerializedType": "Consultingwerk.Framework.MessageInteraction.Question",
    "MessageText": "Der Versicherte ist am 24\/12\/50 verstorben. \n\n
                    Die zugehörigen Wohnadressen werden gesperrt.\n
                    Überprüfen Sie, ob noch Revisionen vorgesehen sind\n
                    und\/oder Hilfsmittel zurückgenommen werden müssen.\n",
    "MessageButtons": "YesNo",
    "MessageReply": "Unanswered",
    "DefaultReply": "ReplyYes",
    "MessageID": "eb09af84b1e2197b:4cb274e8:15608162bb6:-8000",
    "MessageContext": "ac54bf82-56c4-bab2-2514-8e3d5c34775d"
}
```

Questions

Consultingwerk

software architecture and development