

SmartComponent Library
by **Consultingwerk**
software architecture and development

A group of eleven people, including men and women of various ages, are standing outdoors in front of a modern building with large glass windows. They are all smiling and holding a large, light blue banner that spans across the front of the group. The banner has a subtle grid pattern and features the company name and tagline in white text. The background shows green trees and a clear sky.

Consultingwerk

software architecture and development

Consultingwerk Software Services Ltd.

- 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
- 29 years of Progress experience (V5 ... OE11)
- Active member of the OpenEdge community
- Frequent speaker at OpenEdge related conferences around the world





Agenda

- **Overview**
- Framework Backend Architecture
- OpenEdge GUI for .NET Frontends
- Kendo UI and Angular Frontends
- NativeScript Clients
- Repository based cross platform UI
- Application Modernization



SmartComponent Library

- Developer Framework aimed to increase Developer productivity and flexibility
- Reduce or avoid repeating tasks
 - Tools (code generation and round-trip dev.)
 - Integration with Progress tools (OpenEdge, Telerik, KUIB, BPM, Corticon, DataDirect ...)
 - Application Framework
- Allow integration with existing Applications and Frameworks (3rd party or home grown)
- Allow usage of individual framework components as needed (not all or nothing)

SmartComponent Library

- “Eat your own dog food“
- Most of our tools are written using our own libraries and tools
- Nearly 100% ABL, 100% OpenEdge
- **Full source code** shipped to customers
- Typically weekly releases (40-45 releases/year)
- Technical support and maintenance offerings
- Support for new OpenEdge releases in no longer than 30 days

Application Framework Features

- Authentication, including client-principal support, Authentication Gateway and multi-tenancy
- Flexible authorization (menu, toolbar, business logic, custom tokens, ...), can be used for data-related authorization (e.g. Regions)
- Localization (full UI translation), messages, application data
- Definition based referential integrity
- Batch scheduler (repeating, just once, run now)
- Enhanced Unit testing framework
- ...

Agenda

- Overview
- **Framework Backend Architecture**
- OpenEdge GUI for .NET Frontends
- Kendo UI and Angular Frontends
- NativeScript Clients
- Repository based cross platform UI
- Application Modernization



Backend Overview

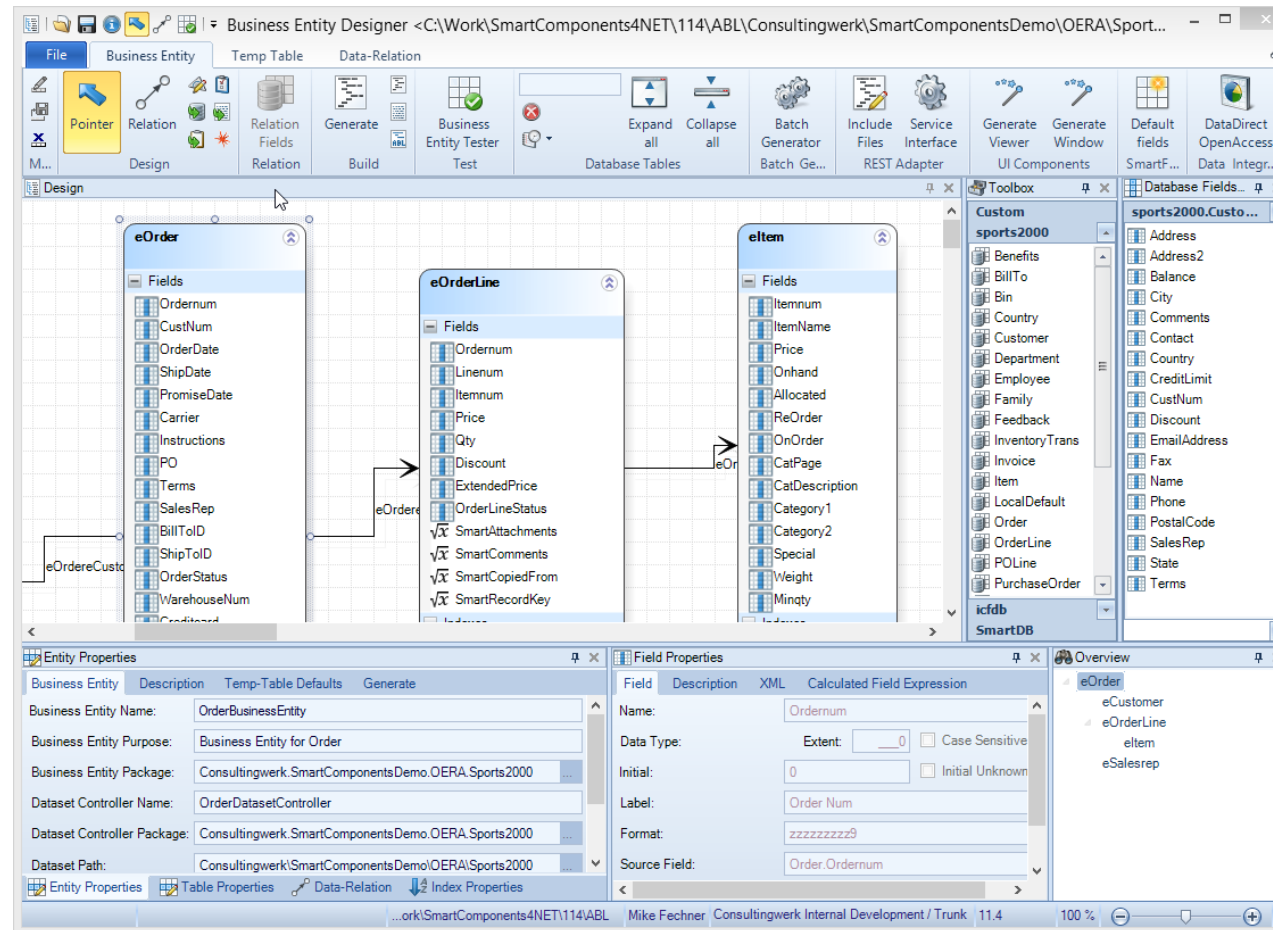
- The SmartComponent Library is based on a flexible OERA Backend Architecture
- OpenEdge Reference Architecture
- Complies with the **Common Component Specification (CCS)**

- Business Entities, Data Access Objects are a key components
- Business Tasks, including support for scheduled and asynchronous processing
- Common Infrastructure Components, Services

Business Logic Design Process

- Template based, Visual Design using Diagrams
- Flexible source code generator
 - Customizable templates
 - Customizable code generator (ABL class)
- Plugin Infrastructure (integrate your own ABL code)
 - Influence automatic naming and design process
 - Add and enhance code generation
- Query custom repositories during design phase
- Ad-hoc based Business Entity Test utility

SmartComponent Library



Demo

- Build Customer Business Entity
 - Validation
 - Calculated Field
- Business Entity Tester
- Use DatasetModel to access Business Entity

Early Sneak Preview: Business Entity Designer on Web

- Re-implement Business Entity Designer UI as Angular Web application
- Leveraging our Angular web components
- Leveraging code generator and validation framework within PASE
- Docker support
- Uploading generated source code to
 - Git
 - Eclipse Che
- Getting ourselves ready for Progress DevOps Hub Developer workspace on the web

RESTful

- Standard RESTful interfaces build into framework
- No custom code required
- Web Developer friendly
- Foundation for API based economy
- Support smart devices like Google Home and Amazon Echo
- Swagger based API documentation



Demo

- Enable RESTful access to Customer Business Entity
- <http://localhost:8820/web/Entities/DemoCustomers>
- Swagger Integration: <http://localhost:8820/web/swagger/entities/html>
- Implement RESTful Business Task method
- <http://localhost:8820/web/Entities/Orders/1/calculateOrderTotal>

SmartFramework ERD Designer

- Based on Zamloxis ERD
- Integration with SmartFramework backend through customized version of Zamloxis ERD
- Simplifies Schema Design and Documentation
- Foundation for definition based referential Integrity
 - Delete restrict / cascade / nullify / set default
 - Validation of foreign key values

11.7 LOCAL - SmartFramework ERD Designer - platform/resource/ABL_117_64/sports2000.smfdia#_u-NAL_3EelBT7hYP3lg - Progress Developer Studio

File Edit Navigate Search Project Run Window Help

Quick Access OpenEdge Editor Planning Debug OpenEdge Server SmartFramework ERD Designer <no task active>

Project Explorer Outline sports2000.smfdia *sports2000.smfdia#Customer

Customer Subject Area

Customer

- CustNum INTEGER
- Name CHARACTER
- Address CHARACTER
- Address2 CHARACTER
- City CHARACTER
- State CHARACTER
- Country CHARACTER
- Phone CHARACTER
- Contact CHARACTER
- SalesRep CHARACTER
- Comments CHARACTER
- CreditLimit DECIMAL
- Balance DECIMAL
- Terms CHARACTER
- Discount INTEGER
- PostalCode CHARACTER
- Fax CHARACTER
- EmailAddress CHARACTER
- CustNum

Salesrep

- RepName CHARACTER
- Region CHARACTER
- SalesRep CHARACTER
- MonthQuota INTEGER
- SmallImage BLOB
- LargeImage BLOB
- Email CHARACTER
- PhoneOffice CHARACTER
- PhonePrivate CHARACTER
- PhoneMobile CHARACTER
- SalesRep

Order

- Ordernum INTEGER
- CustNum INTEGER
- OrderDate DATE
- ShipDate DATE
- PromiseDate DATE
- Carrier CHARACTER

fk_Salesrep_Customer 1..*

fk_Salesrep_Order 1..*

fk_Customer_Order 1..*

Table

Property	Value
Label	
Replication	
Triggers	CREATE: sports2000trgs/crcust.p, DELETE
SmartFramework	
Allow Attachments	false
Allow Comments	false
Description Fields	CustNum
Description Fields Substitute	
Key Field Assignment	
Temp Table Name	
Unique Fields	

Related Tables View

Name	Parent	Child	Cardinality	Field map
fk_Country_Customer	Country	Customer	1..*	[Country -> Country]
fk_Customer_BillTo	Customer	BillTo	1..*	[CustNum -> CustNum]
fk_Customer_Invoice	Customer	Invoice	1..*	[CustNum -> CustNum]
fk_Customer_Order	Customer	Order	1..*	[CustNum -> CustNum]
fk_Customer_RefCall	Customer	RefCall	1..*	[CustNum -> CustNum]
fk_Customer_ShipTo	Customer	ShipTo	1..*	[CustNum -> CustNum]
fk_Salesrep_Customer	Salesrep	Customer	1..*	[SalesRep -> SalesRep]
fk_State_Customer	State	Customer	1..*	[State -> State]

358M of 781M

Zur Suche Text hier eingeben

20:12 02.11.2017

Unit Testing

- Framework supports test-driven development in multiple ways
- Business Logic can be tested against “Mock” of the Database or individual data-source components
- Perform unit tests that perform independent from altered data in the database
- Support mocking of dependent code through service manager or configurable object instance factory
- Scenario driven unit tests for business entity read and update operations
 - Eliminate need to write similar unit tests for all your components

Agenda

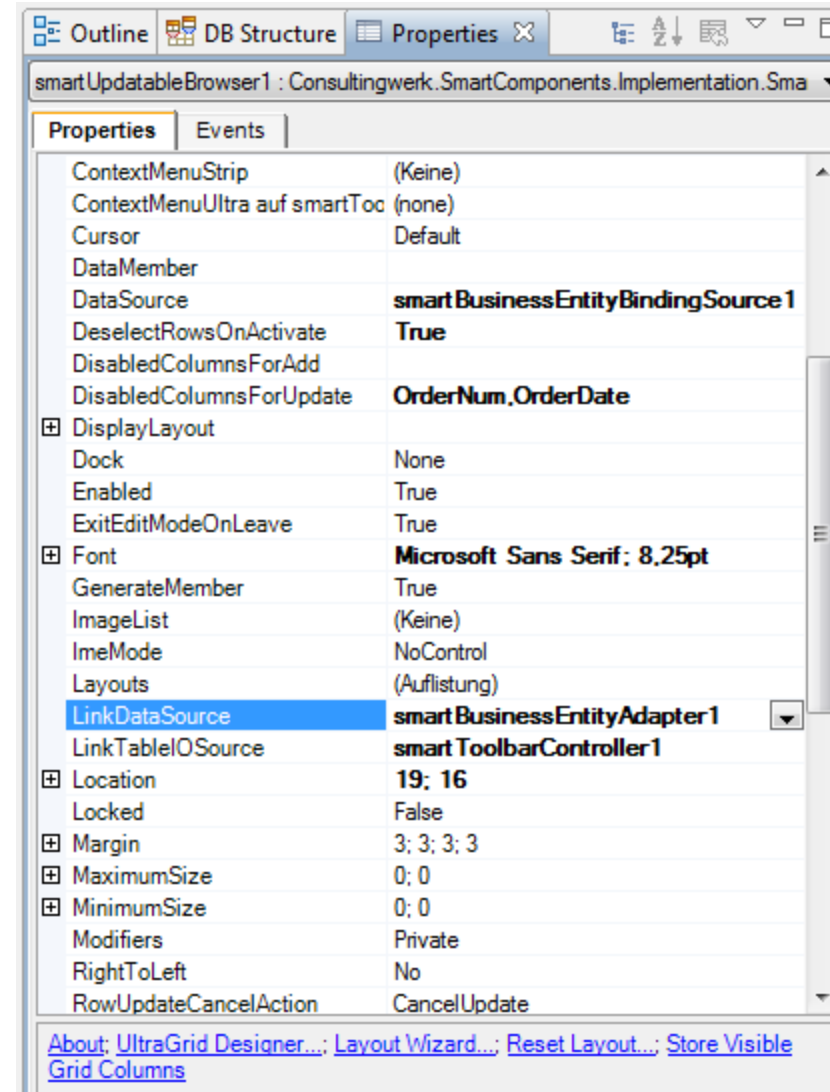
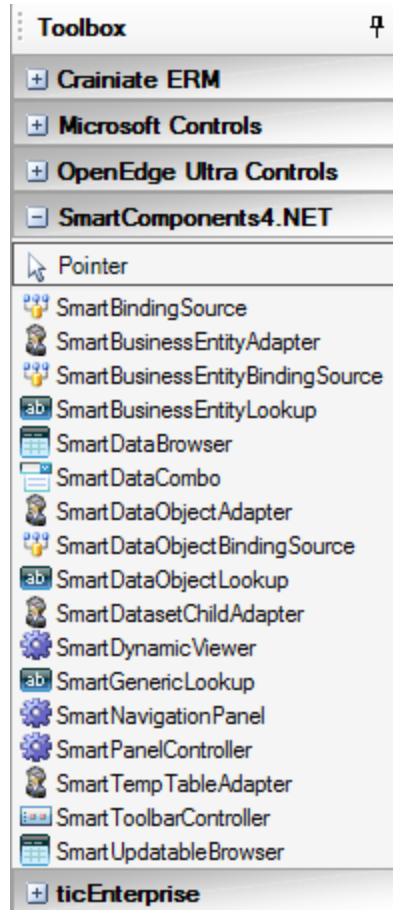
- Overview
- Framework Backend Architecture
- **OpenEdge GUI for .NET Frontends**
- Kendo UI and Angular Frontends
- NativeScript Clients
- Repository based cross platform UI
- Application Modernization



User Interface Design Process

- Screen design based on rich foundation classes
- Compose Forms by adding rich SmartComponents and standard .NET Controls
- Connect and parameterize SmartComponents using property grid and Wizards
- Data centric design process: Achieving UI Design productivity similar to the AppBuilder
- Screen design either in Progress Developer Studio Visual Designer or our repository based tooling

User Interface Design Process



Demo

- Use Business Entity Designer to generate UI
- Open in Visual Designer
- Add Salesrep Lookup in Visual Designer

Agenda

- Overview
- Framework Backend Architecture
- OpenEdge GUI for .NET Frontends
- **Kendo UI and Angular Frontends**
- NativeScript Clients
- Repository based cross platform UI
- Application Modernization



Support for JSDO Based Clients

- SmartComponent Library Backend and

JSDO based consumers: Telerik Mobile, Kendo UI, Kendo UI Builder, ... = perfect match!

- Full support for complete JSDO functionality

JSDO  **Consultingwerk likes this.**

Support for Progress/Telerik Mobile

- Generic REST Service Interface for JSDO based clients
 - Full CRUD support
 - Invoke Business Entity Methods and Business Tasks
- No need to redeploy new REST application for every single application
- Direct access to any Business Entity or Business Task that the current user is authorized for
- Integrated Localization, translation of field labels etc.
- Simplified deployment (small and large teams)
- PASOE (Progress AppServer) or Classic AppServer

Recorded Demo Telerik Mobile Platform

- Create a new hybrid mobile project in Telerik Mobile platform using Wizard
- Add access to SalesRep Business Entity
- Leverage same Business Entity in Web and Desktop
- Recorded version of the demo: <https://www.youtube.com/watch?v=m-63Wzpjs9c>

Support for Kendo UI Builder

- SmartComponent Library can be used as the backend for KUIB
- SmartComponent Library can be used as the backend for every JSDO based application (e.g. Kendo UI)

SmartComponents.Web2

- Our own Angular Client



Kendo UI
THE ART OF WEB DEVELOPMENT



SmartComponents.Web2

- Angular is a MVVM framework for browser based applications
- Binding Data to UI Elements
- Separate Data, View and Code
- Kendo UI provides UI components for Angular
- SmartComponents.Web2 implements similar programming model as in GUI for .NET
- Configurable data sources and communication between UI components

Multiple screen design options

- HTML based Angular templates, referencing our components
- JSON based layout, abstracted, supports re-use in NativeScript
- Annotations, simple layouts rendered on the fly
- User Interface Repository, supporting **reuse** of UI designs (e.g. Viewers, Grids) on **Desktop, Web and Mobile**

Screen Definition based on functional components

```
<div ng-controller="simpleCtrl" style="background-color: white;">
  <h1>Salesreps</h1>

  <div class="row">
    <div class="col-md-28">

      <smart-data-source
        smart-object-name='salesrepDataSource'
        smart-business-entity-name="Consultingwerk.SmartComponentsDemo.OERA.Sports2000.SalesRepBusinessEntity"
        smart-entity-table="eSalesrep"></smart-data-source>
      <smart-grid
        smart-object-name='salesrepGrid'
        smart-data-source='salesrepDataSource'
        smart-grid-layout='Consultingwerk.SmartComponentsDemo.OERA.Sports2000.SalesRepBusinessEntity/salesrep'></smart-grid>

    </div>

    <div class="col-md-20">

      <smart-toolbar
        smart-object-name='salesrepToolbar'></smart-toolbar>

      <smart-viewer
        smart-object-name='salesrepViewer'
        smart-data-source='salesrepDataSource'
        smart-tableio-source='salesrepToolbar'
        smart-viewer-layout='Consultingwerk.SmartComponentsDemo.OERA.Sports2000.SalesRepBusinessEntity/salesrep'></smart-viewer>

    </div>
  </div>
</div>
```



Demo

- SmartComponents.NG2 sample application
- Screen definition in Repository Tooling



Kendo UI
THE ART OF WEB DEVELOPMENT

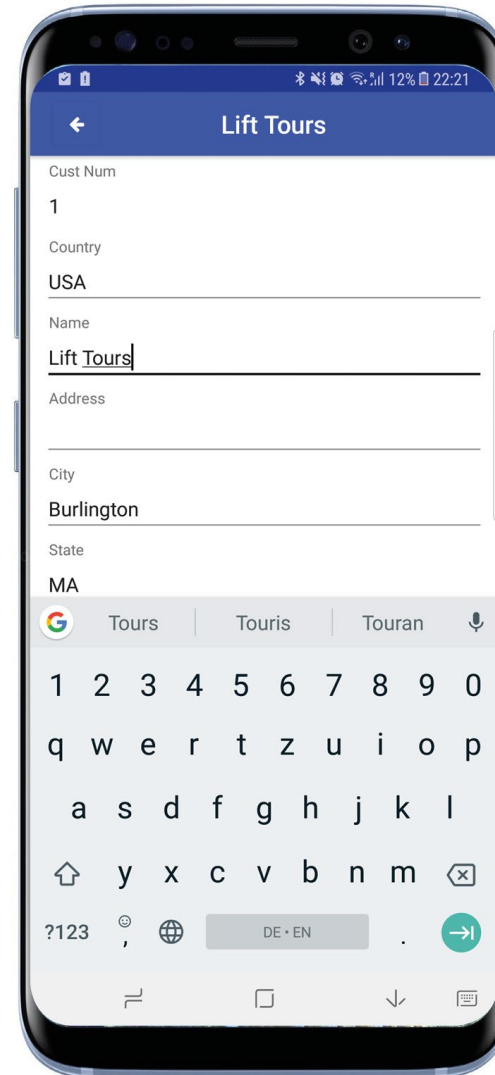
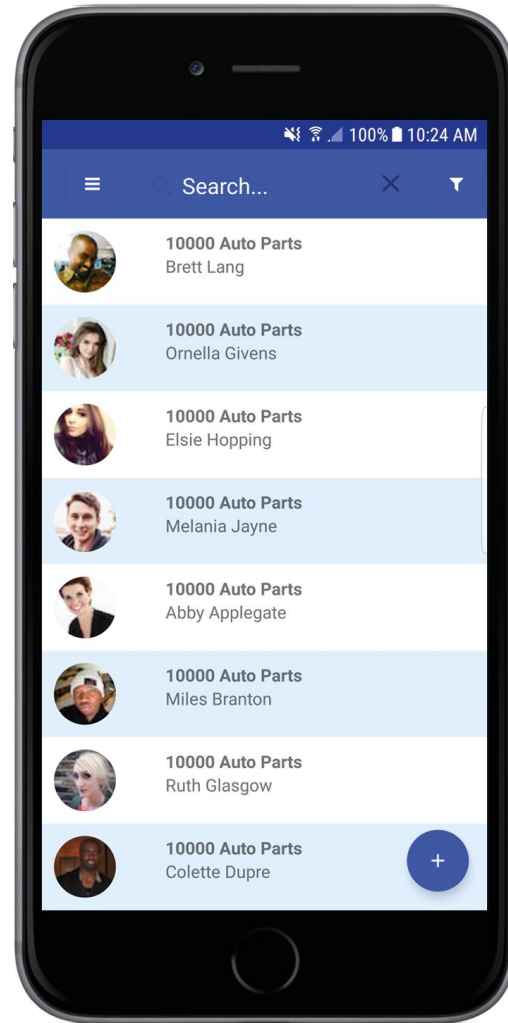
Agenda

- Overview
- Framework Backend Architecture
- OpenEdge GUI for .NET Frontends
- Kendo UI and Angular Frontends
- **NativeScript Clients**
- Repository based cross platform UI
- Application Modernization



NativeScript

- Native mobile applications for iOS and Android
- JavaScript or TypeScript based
- Standard for Angular based mobile applications
- Allows us to reuse components of SmartComponents.NG2 on mobile
- Allows developers to reuse client side code
- Similar developer experience for customers
- Similar component architecture
 - Data sources, grids, viewer, lookups, ...



Kinvey mBaaS Adapter

- SmartComponents for NativeScript include a Kinvey Adapter
- Leverage cloud based hosted backend services
- Support caching, asynchronous updates and conflict resolution
- Based on JSDO support in the NativeScript and Kinvey

Agenda

- Overview
- Framework Backend Architecture
- OpenEdge GUI for .NET Frontends
- Kendo UI and Angular Frontends
- NativeScript Clients
- **Repository based cross platform UI**
- Application Modernization



Repository based UI

- SmartFramework supports user interface repository
- Leveraging the same user interface layout on multiple devices (desktop, web and mobile)
- UI versioning with code (using Git, Perforce, Roundtable, ...)
- Mixed applications supported
 - repository based UI for standard screens
 - static screen layout for specialized screens
- Bridge responsive and pixel-perfect UI design
 - Desktop typically pixel perfect
 - Web, mobile typically responsive

Demo

- Repository based UI Designer
- Desktop and Web (early Sneak Preview)

Agenda

- Overview
- Framework Backend Architecture
- OpenEdge GUI for .NET Frontends
- Kendo UI and Angular Frontends
- NativeScript Clients
- Repository based cross platform UI
- **Application Modernization**



SmartComponent Library and WinKit

- ***SmartComponent Library*** and ***WinKit*** are part of our GUI for .NET adoption and migration strategy
 - WinKit: Enhancement of existing screens
 - SmartComponent Library: New functionality
- Shared framework foundation classes
- Compatible directory and project structure
- Used side by side in customer projects
- Over 45 different WinKit implementations already
- WinKit included in SmartComponent Library

Order Maintenance

File Navigate Maintenance View

Order Num: 1 Ordered: 23/01/09

Order Status: Shipped

Shipped: 28/01/2009

Promised: 28/01/09

Cust Num: 53 Name: Offside Hockey

Carrier: FlyByNight Courier Terms: Net30 PO:

Orderlines Customer Detail

Handle with care

OrderLines

Line Num	Item Num	Item Name	Qty	Price	Extende
1	54	Canoe	30	4,86	
2	55	Surfboard	21	23,85	
3	53	Swimming Trunks	75	8,77	
4	28	Ski Gloves	39	23,00	
5	39	Baseball	75	34,00	2.
6	26	Catchers Mit	28	4,99	

Line Num: 1

Item Num: 54

Item Name: Canoe

Price: 4,86

Qty: 30 Extended Price: 131,22

Discount: 10%

Order Line Status: Shipped

Order Maintenance - Redesigned

File Start Options

Navigation: First, Prev, Next, Last
 Search: Search
 Maintenance: Edit, Save, Delete, Cancel

Order Num: 1 Ordered: 23/01/09

Order Status: Shipped

Shipped: 28/01/2009
 Promised: 28/01/09

Cust Num: 53 Name: Offside Hockey

Carrier: FlyByNight Courier Terms: Net30 PO:

Handle with care

Orderlines Customer Detail

Legen Sie eine Spalte hier ab, um nach dieser zu gruppieren.

Line Num	Item Num	Item Name	Qty	Price	Exte
1	54	Canoe	30	4,86	
2	55	Surfboard	21	23,85	
3	53	Swimming Trunks	75	8,77	
4	28	Ski Gloves	39	23,00	
5	39	Baseball	75	34,00	
6	26	Catchers Mit	28	4,99	

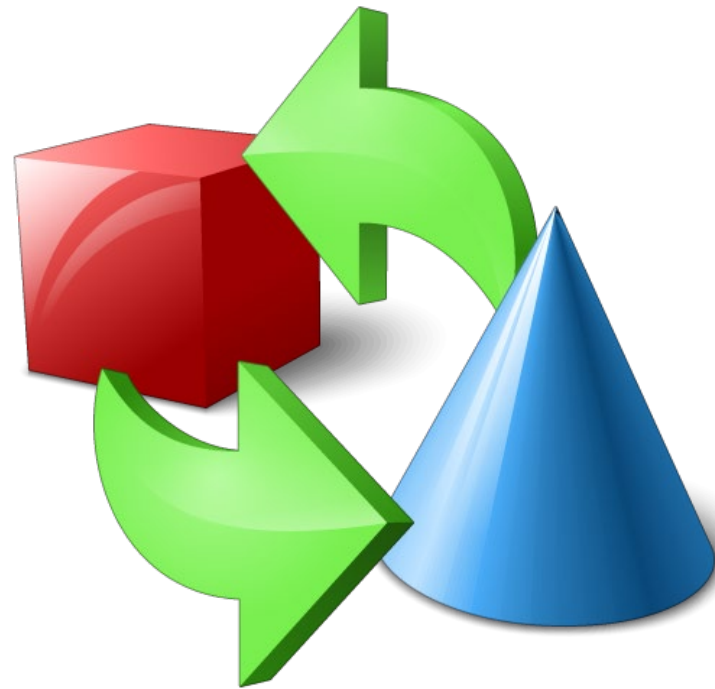
Line Num: 1
 Item Num: 54
 Item Name: Canoe
 Price: 4,86
 Qty: 30 Extended Price: 131,22
 Discount: 10%
 Order Line Status: Shipped

Migration and Modernization options

- **Enhance and extend existing applications**
- SmartComponent Library integrates with existing applications and frameworks
- SmartComponent Library functionality can be accessed from existing code
- SmartComponent Library can access existing code, typically of any kind
 - direct execution or through adapters
 - custom service or component implementations
 - Don't worry about SHARED VARIABLES etc...

Alternative

- Migration of existing Applications



Migration and Modernization options

- Semi-automated Migration of source code
- Options vary based on current code
- Strong usage of Proparse for analyzing source code, importing into Business Entity Designer
- Refactoring tooling based on Proparse
- Migration of any kind of applications
 - custom
 - ADM1, ADM2 or Dynamics
- Template based migration of UI components
- API for ABL Widget abstraction

Demo

- Migration of ABL Browse Widget
 - Rewrite of Calculated Field
 - Creation of matching Business Entity
- Migration of ABL GUI for .NET Frame
 - Implement ABL Widget abstraction

Questions

- Email us at info@consultingwerk.com

