Black-Belt Development

Part 2

Sergey Marenich
Solution Architect
Acumatica
sm@acumatica.com | http://asiablog.acumatica.com

Sergey Marenich

Experience: 11 years at Acumatica

- 7 years as a system developer in R&D center
- 4 years as a technical consultant & architect in South East Asia region

Modules developed previously:

- Installer, Configuration Wizard, Licenses
- Files Attaching & Sync, Companies
- Snapshots Field Level Audit, Localization
- Exchange Integration, Online Updated
- Single Sign On, Generic Inquires
- Users and Security, Integration Services

Agenda

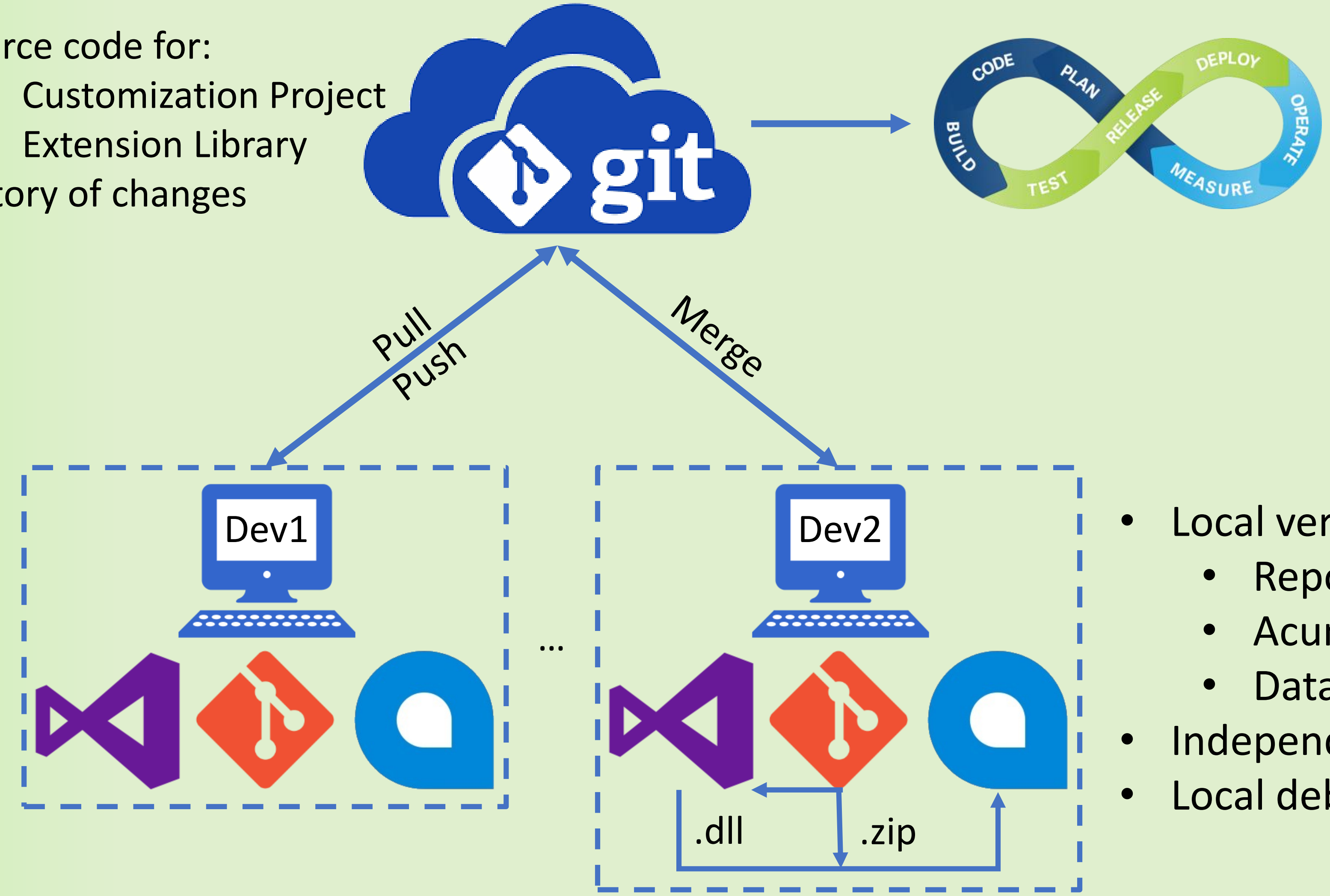
- Acumatica Platform for Teamwork
  - Team Development
  - Test Framework
  - Continuous Integration
- Acumatica Development Patterns
  - 6 Patterns

TEAM DEVELOPMENT

Acumatica
Developer Network

Team Development

- Source code for:
  - Customization Project
  - Extension Library
- History of changes



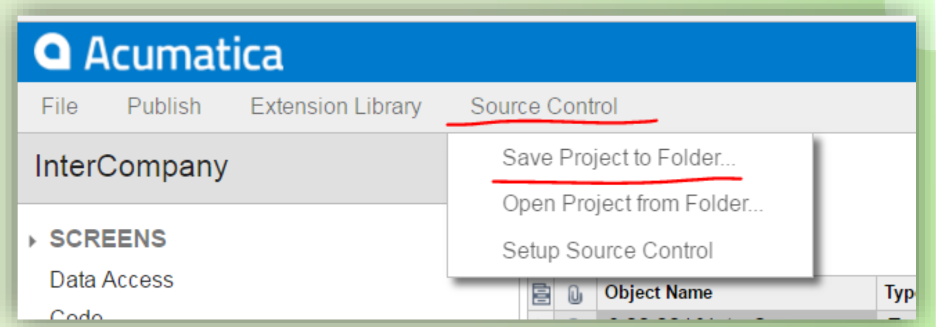
- Build
- Test
- Deploy

- Local version of:
  - Repository
  - Acumatica Instance
  - Database
- Independent development
- Local debug

Acumatica
The Cloud ERP

Team Development – Example

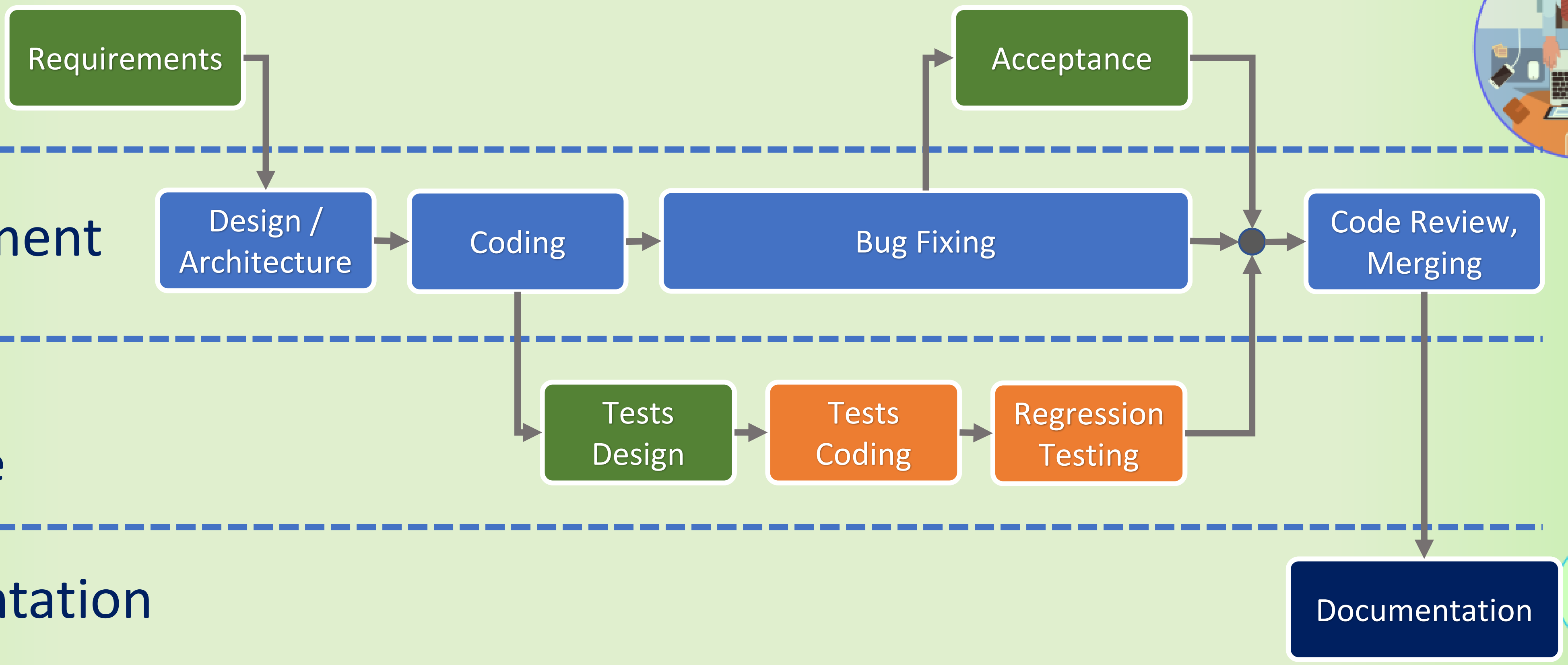
- Setup Local Environment
  - Download project from Source Control
  - Install Acumatica into the <u>Predefined</u> place
  - Compile local DLL project
  - Create Customization project in Acumatica
  - Open project using Source Control
- Do Development
- Commit Changes
  - Save project changes to folder
  - Commit and Push changes in Source Control
- https://github.com/Acumatica/CustomizationBuildScript



Team Development – Lifecycle

🟩 Product Management 🟦 Development 🟧 QA Engineer 🟦 Technical Writer

- Analysis & Design
- Development
- Quality Assurance
- Documentation



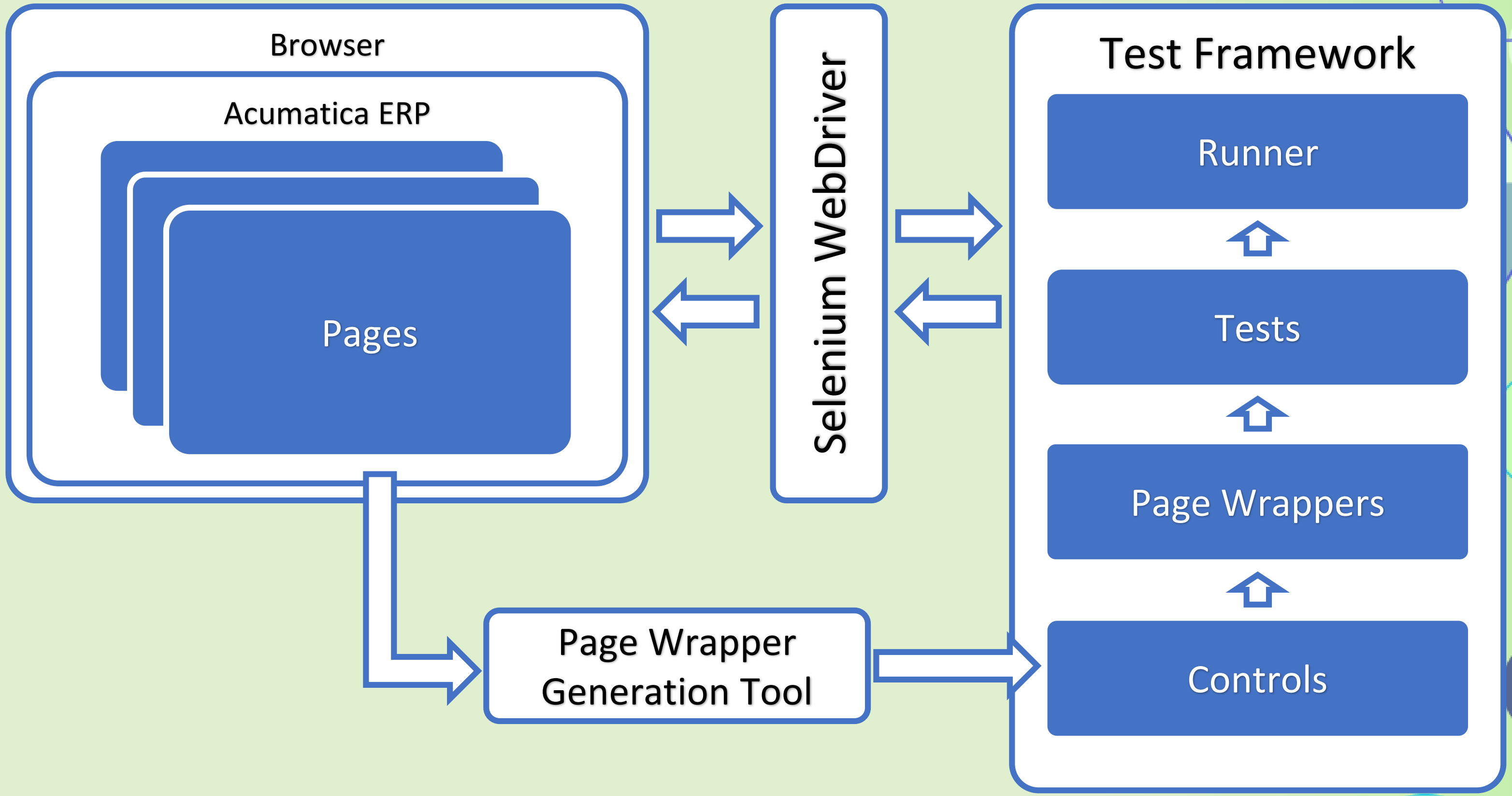
Acumatica
The Cloud ERP

TEST FRAMEWORK

Acumatica
Developer Network

Test Framework

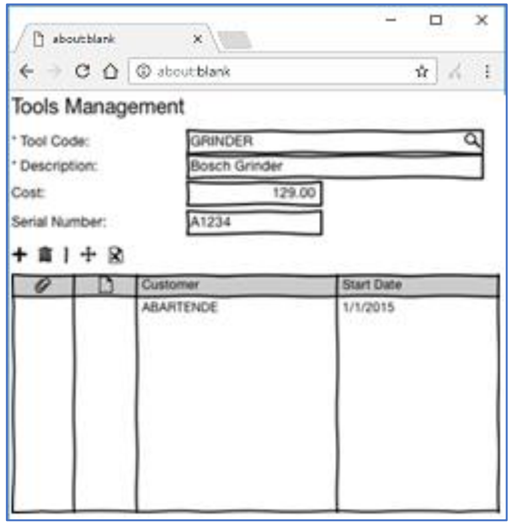
- Black Box Testing
  - Browser UI
  - Mobile UI
- Selenium Web Driver
- You own reusable regression test
- Test Results



Acumatica
The Cloud ERP

Test Framework – How does it work?

Browser


about:blank

Tools Management

* Tool Code:	GRINDER
* Description:	Bosch Grinder
Cost:	129.00
Serial Number:	A1234

		Customer	Start Date
		ABARTENDE	1/1/2015

```
<div class="cell-pv">
 <div class="cell-w">
  <div class="fld-l fld-ls">
   <label for="ctl00_phF_form_edToolCD_text">Tool Code:</label>
  </div><div class="fld-c fld-cs">
  <div name="ctl00$phF$form$edToolCD"
   id="ctl00_phF_form_edToolCD" class="selector required size-s">
   <div class="controlCont">
    <div class="editorCont">
     <div class="editorWrap">
       <input name="ctl00$phF$form$edToolCD$text"
        type="text" tabindex="18105" class="editor"
        id="ctl00_phF_form_edToolCD_text“ />
     </div>
    </div>
   </div>
  </div>
 </div>
</div>
```

Page Wrapper Generator Tool

Code Generated

```
public class ToolManagement : Wrapper
{
  public summary_form form = new form("ctl00_phF_form");
  public class summary_form : Container
  {
    public Selector ToolCD { get; set; }
    public summary_form(string locator)
    {
      …
      ToolCD = new Selector(
        "ctl00_phF_form_edToolCD_text",
       “Tool Code", locator, null);
      ToolCD.DataField = “ToolCD";
}
  }
  public virtual void Save() { … }
}
```

User Interface

Browser API

"ctl00_phF_form_edToolCD_text"

Test Scenario

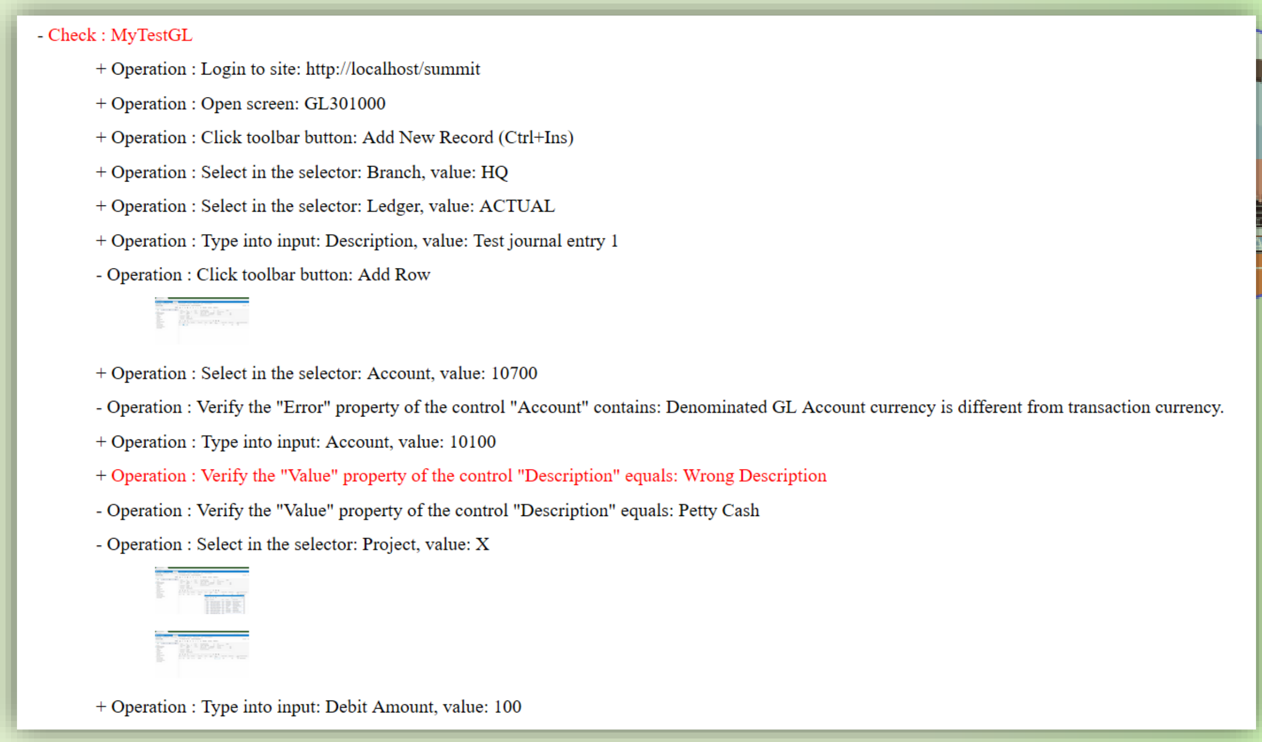
```
public class MyTestGLBob : Check
{
  public override void Execute()
  {
    var toolsForm = new ToolManagement();
    form.OpenScreen();

    form.Summary.ToolCD.Type("ABC");
    form.Summary.Description.Type("My new tool");
    form.Save();
  }
}
```

Acumatica
The Cloud ERP

Test Framework – Example

- Get Test Framework
  - http://builds.Acumatica.com
  - Check “SDK - README.pdf”
- Generate Wrappers
- Write Own Test
- Run Test
- Collect Results



```
- Check : MyTestGL
    + Operation : Login to site: http://localhost/summit
    + Operation : Open screen: GL301000
    + Operation : Click toolbar button: Add New Record (Ctrl+Ins)
    + Operation : Select in the selector: Branch, value: HQ
    + Operation : Select in the selector: Ledger, value: ACTUAL
    + Operation : Type into input: Description, value: Test journal entry 1
    - Operation : Click toolbar button: Add Row

    + Operation : Select in the selector: Account, value: 10700
    - Operation : Verify the "Error" property of the control "Account" contains: Denominated GL Account currency is different from transaction currency.
    + Operation : Type into input: Account, value: 10100
    + Operation : Verify the "Value" property of the control "Description" equals: Wrong Description
    - Operation : Verify the "Value" property of the control "Description" equals: Petty Cash
    - Operation : Select in the selector: Project, value: X

    + Operation : Type into input: Debit Amount, value: 100
```

Acumatica
The Cloud ERP

CONTINUOUS INTEGRATION

Acumatica
Developer Network

Continuous Integration – Part 1

- Automate Acumatica Wizard Installation
  - msiexec.exe /i /a /qn “AcumaticaERPInstall.msi"
- Automate Acumatica Instance Deployment
  - ac.exe -cm:"NewInstance" -s:"*<SqlSrv>*" -d:"AcuInst" -i:"AcuInst" -h:"<Path>" -w:"Default Web Site" -v:"AcumaticaERP" -po:"DefaultAppPool“
- Getting Repository
  - $ git clone https://github.com/Acumatica/CustomizationBuildScript.git
- Automate Compilation
  - MSBuild Cust.sln /property:Configuration=Release /t:Rebuild

Acumatica
The Cloud ERP

Continuous Integration – Part 2

- Automate Build of Customization Project
  - PX.CommandLine.exe /method BuildProject /website "<AcumaticaSitePath>" /in "<CustProjectSources>" /out "Cust.zip"
- Automate Customization Publication
  - https://<instance>/api/servicegate.asmx
    - acu.UploadPackage("Cust", File.ReadAllBytes("Cust.zip"), true);
    - acu.PublishPackages(new string[] { "Cust" }, false);
- Automate Tests Launch
  - ClassGenerator.exe
  - MyTest.exe /config "example.xml"

Acumatica
The Cloud ERP

DEVELOPMENT PATTERNS

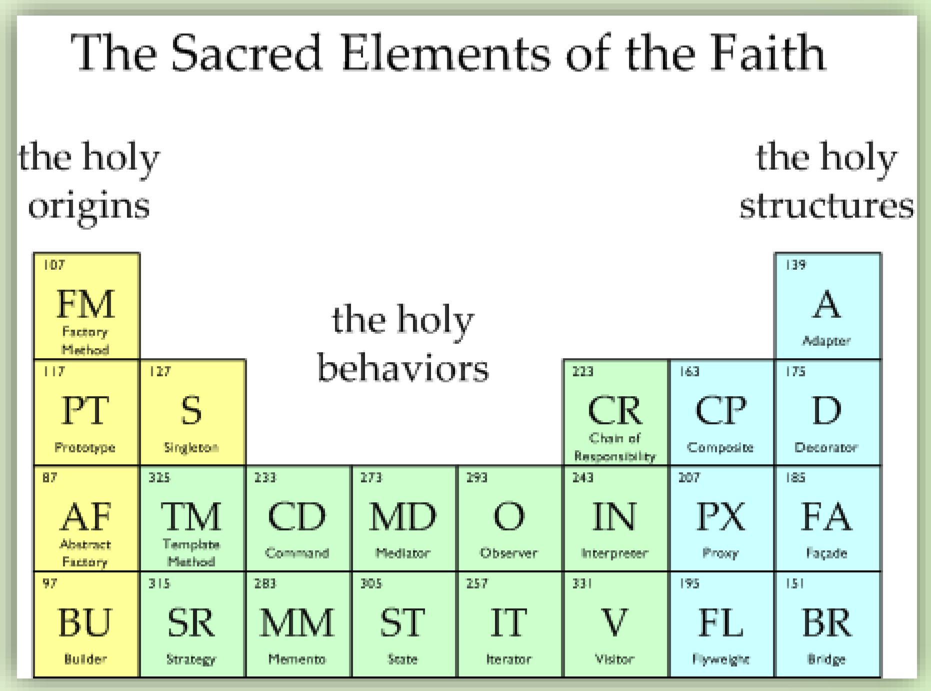
Acumatica
Developer Network

Development Patterns

Patterns - Reusable solution to a commonly occurring problems

Structure:

- Problem Statement
- Resolution



The Sacred Elements of the Faith

the holy origins

the holy behaviors

the holy structures

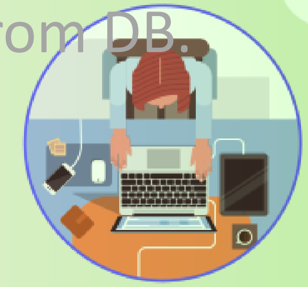
107 FM Factory Method							139 A Adapter
117 PT Prototype	127 S Singleton				223 CR Chain of Responsibility	163 CP Composite	175 D Decorator
87 AF Abstract Factory	325 TM Template Method	233 CD Command	273 MD Mediator	293 O Observer	243 IN Interpreter	207 PX Proxy	185 FA Façade
97 BU Builder	315 SR Strategy	283 MM Memento	305 ST State	257 IT Iterator	331 V Visitor	195 FL Flyweight	151 BR Bridge

Totals Calculation for Inquiry

Development Patterns

Totals Calculation for Inquiry - Problem

It is not really good idea to iterate thought all records if you need to get a total of records selected, as you have to ignore paging and have to select all records from DB.



```
public PXFilter<InquiryFilter> Filter;
public PXSelect<Batch> Records;
public IEnumerable records()
{
    decimal docsTotal = 0m;
    foreach(Batch row in Records.Select())
    {
        docsTotal += res.ControlTotal;
        yield return row;
    }
    //...
}
```

Acumatica
The Cloud ERP

Totals Calculation for Inquiry - Solution

Calculate totals in separate select with aggregation and than select records with paging as usual.

```
public PXFilter<InquiryFilter> Filter;
public PXSelect<Batch> Records;
public IEnumerable filter()
{
    Batch row = PXSelect<Batch,
        Aggregate<GroupBy<Batch.batchType,
        Sum<Batch.controlTotal>>>>>.Select();
    decimal docsTotal = row.ControlTotal;
    //...
}
```

Acumatica
The Cloud ERP

Attributes Duplication

Development Patterns

Attribute Duplication - Problem

Common fields need to duplicate attributes in all DACs.

```
public class ARInvoice : IBqlTable
{
	[PXDBInt()]
	[PXUIField(DisplayName = "Customer")]
	[PXDefault()]
	[PXSelector(typeof(Search<BAccountR.bAccountID,
		Where<BAccountR.status, Equal<BAccount.status.active>>>))]
	public override Int32? CustomerID
}
public class SOOrder : IBqlTable
{
	[PXDBInt()]
	[PXUIField(DisplayName = "Customer")]
	[PXDefault()]
	[PXSelector(typeof(Search<BAccountR.bAccountID,
		Where<BAccountR.status, Equal<BAccount.status.active>>>))]
	public override Int32? CustomerID
}
```

Acumatica
The Cloud ERP

Attribute Duplication - Solution

- Usage of PXAggregate Attribute to combine common attributes in one.

```
[PXDBInt()]
[PXUIField(DisplayName = "Customer")]
[PXDefault()]
[PXSelector(typeof(Search<BAccountR.bAccountID,
    Where<BAccountR.status, Equal<BAccount.status.active>>>))]
public class CustomerActiveAttribute : PXAggregateAttribute { }
```

-
  ```
  public class ARInvoice : IBqlTable
  {
      [CustomerActive]
      public override Int32? CustomerID { get; set;}
  }
  ```

Acumatica
The Cloud ERP

Multiple Calls of DataView Delegate

Development Patterns

Multiple Calls of DataView Delegate - Problem

- If you work with virtual DAC that is constructed in DataView you may need to construct it multiple times as DataView delegate can be called multiple times. That affects performance

```
public PXSelect<BigVirtualDAC> Records;
public IEnumerable records()
{
    BigVirtualDAC row = new BigVirtualDAC();
    yield return row;
}
```

Acumatica
The Cloud ERP

Multiple Calls of DataView Delegate - Solution

Cache.Insert() will help you to store constructed records between calls. Records will be stored in the session.

```
public PXSelect<BigVirtualDAC > Records;
public IEnumerable records()
{
        Boolean anyfound = false;
        foreach(BigVirtualDAC row in OpenDocuments.Cache.Inserted)
        {
                anyfound = true;
                yield return row;
        }

        if (!anyfound)
        {
                BigVirtualDAC row = new BigVirtualDAC();
                row = Records.Insert(row);

                yield return row;
        }
}
```

Acumatica
The Cloud ERP

Passing Parameters to Processing Method

Development Patterns

Passing Parameters to Processing Method - Problem

Anonymous delegate can help you to pass additional parameters into the processing delegate.

Please note that you can pass there only local variables to eliminate reference to graph. Otherwise you will have synchronous processing

```
public PXFilteredProcessing<ARRegister, Filter> ProcessingView

public ProcessingGraph()
{
    ProcessingView.SetProcessDelegate(ProcessDocuments);
}

public static void ProcessDocuments(List<ARRegister> documents)
{
}
```

Acumatica
The Cloud ERP

Passing Parameters to Processing Method - Problem

Cache.Insert() will help you to store constructed records between calls. Records will be stored in the session.

Can do the same on RowSelected event instead of constructor.

```
public PXFilteredProcessing<ARRegister, Filter> ProcessingView
public ProcessingGraph()
{
      Filter filter = FilterView.Current;
      DetailsView.SetProcessDelegate(delegate (List<ARRegister> documents)
      {
            ProcessDocuments(filter, documents);
      });
}

public static void ProcessDocuments(Filter filter, List<ARRegister> documents)
{
}
```

Acumatica
The Cloud ERP

Dynamic BQL

Development Patterns

Dynamic BQL - Problem

- Construction of BQL Generics Dynamically is complicated. Even with Compose method is hardly readable and not easy to use.

```
Type bql = BqlCommand.Compose(
    typeof(Search<,>), typeof(INUnit.fromUnit),
        typeof(Where<,,>), typeof(INUnit.unitType),
            typeof(Equal<INUnitType.global>),
    typeof(And<,>), typeof(INUnit.toUnit),
        typeof(Current<>), BaseUnitType);

var selectorAttribute = new PXSelectorAttribute(bql);
```

Acumatica
The Cloud ERP

Dynamic BQL - Solution

- Usage of BQL Templates to easily replace placeholder part with sub BQL query.
  BqlTemplate<>.ToType() is also supported

-
  ```
  BqlCommand command = BqlTemplate<Search<INUnit.fromUnit,
          Where<INUnit.unitType, Equal<INUnitType.global>,
              And<INUnit.toUnit,
                  Equal<Current<BqlPlaceholder1>>>>>>
      .Replace<BqlPlaceholder1>(BaseUnitType).ToCommand();

  PXView view = new PXView(this, true, command);
  ```

Acumatica
The Cloud ERP

Reusable Business Objects

Development Patterns

Reusable Business Objects - Problem

It is a nightmare when you have complex business logic that involves multiple tables and you need to use it across of the multiple screens.

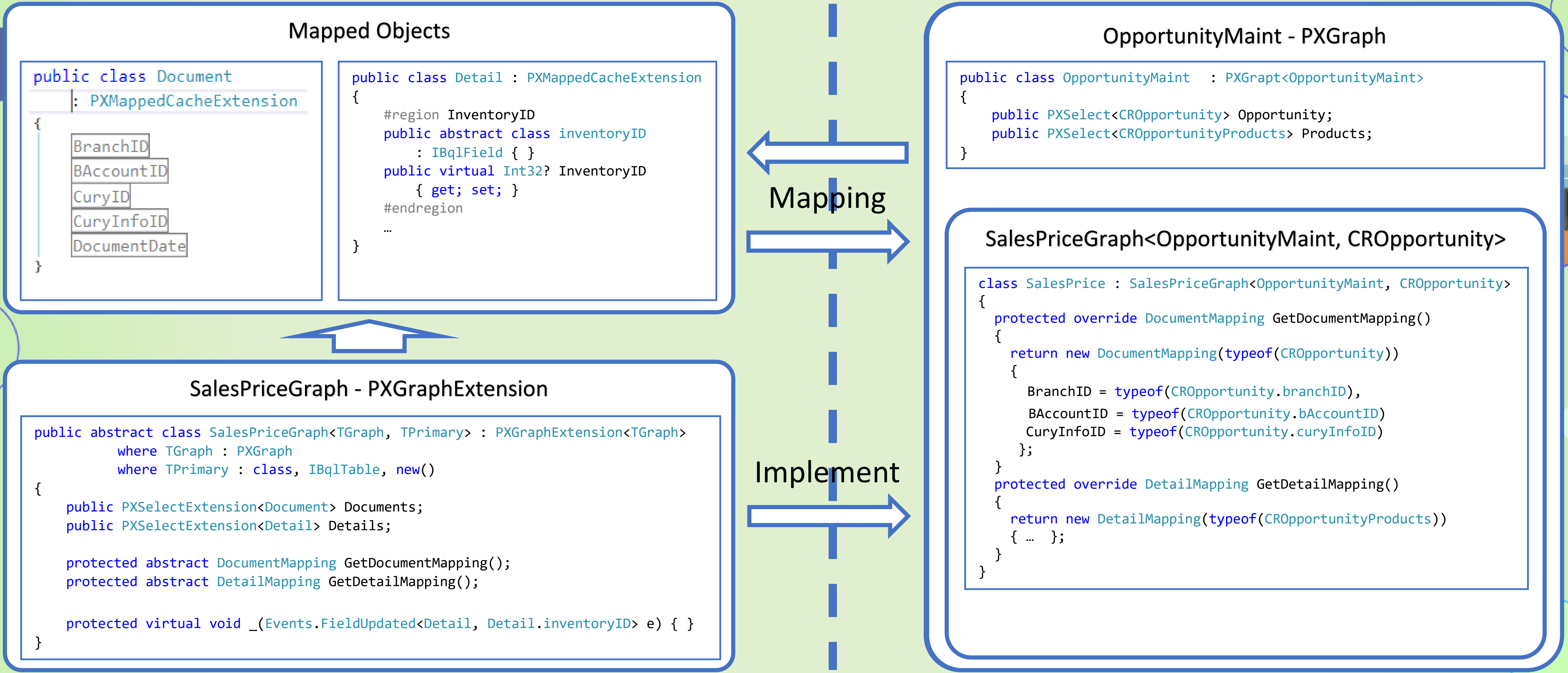
Options:

- Use Attributes to share logic (Multi-Currency Attribute, TaxAttribute)
- Use PXSelect to embed logic there (ApprovalAutomation, LSSelect)
- Copy Duplication

Issues: ~~not possible~~ complicated to reuse it, support it.

Acumatica
The Cloud ERP

Reusable Business Objects - Solution



Mapped Objects

```
public class Document
    : PXMappedCacheExtension
{
    BranchID
    BAccountID
    CuryID
    CuryInfoID
    DocumentDate
}
```

```
public class Detail : PXMappedCacheExtension
{
    #region InventoryID
    public abstract class inventoryID
        : IBqlField { }
    public virtual Int32? InventoryID
        { get; set; }
    #endregion
    …
}
```

SalesPriceGraph - PXGraphExtension

```
public abstract class SalesPriceGraph<TGraph, TPrimary> : PXGraphExtension<TGraph>
          where TGraph : PXGraph
          where TPrimary : class, IBqlTable, new()
{
    public PXSelectExtension<Document> Documents;
    public PXSelectExtension<Detail> Details;

    protected abstract DocumentMapping GetDocumentMapping();
    protected abstract DetailMapping GetDetailMapping();

    protected virtual void _(Events.FieldUpdated<Detail, Detail.inventoryID> e) { }
}
```

OpportunityMaint - PXGraph

```
public class OpportunityMaint  : PXGrapt<OpportunityMaint>
{
    public PXSelect<CROpportunity> Opportunity;
    public PXSelect<CROpportunityProducts> Products;
}
```

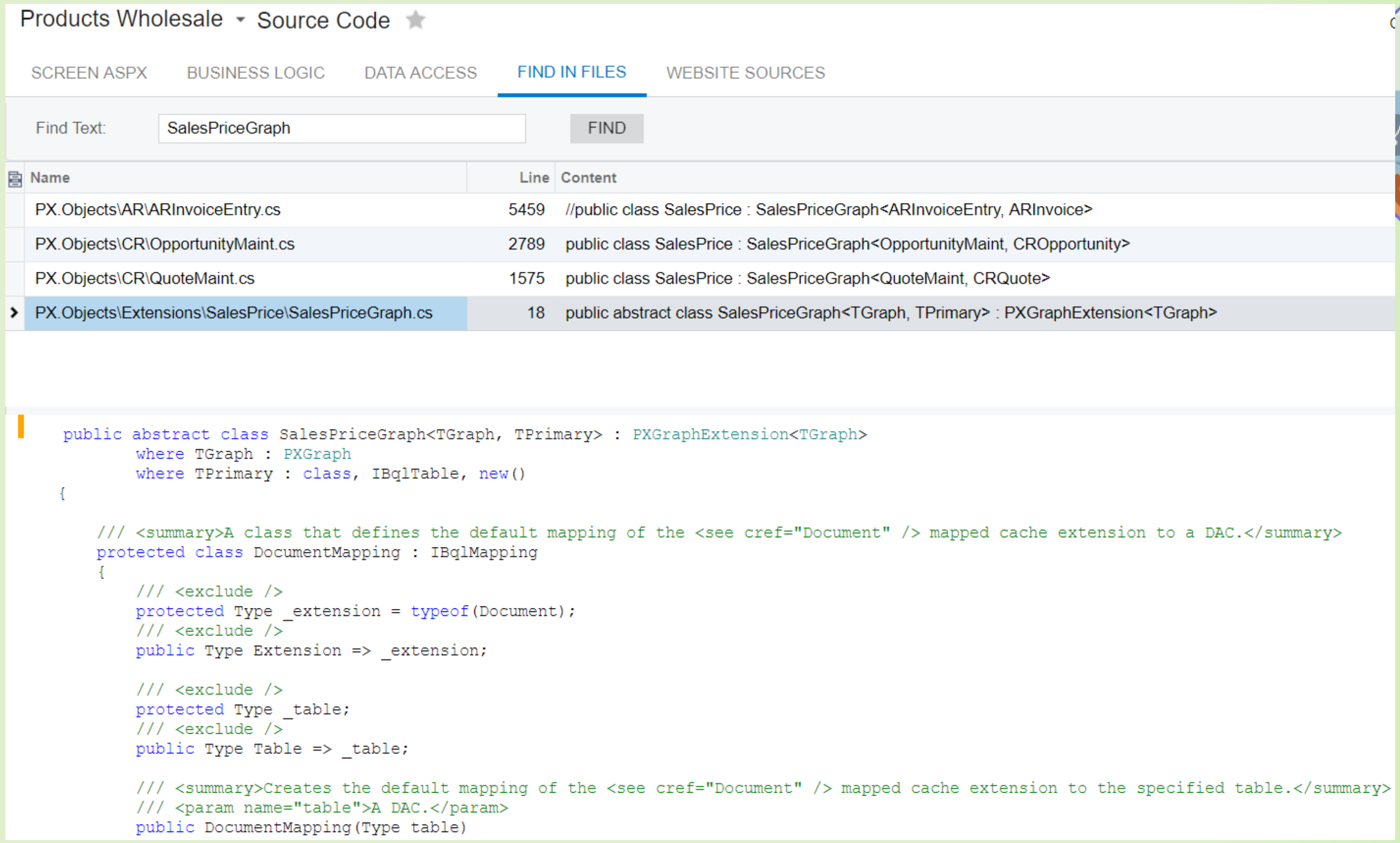
SalesPriceGraph<OpportunityMaint, CROpportunity>

```
class SalesPrice : SalesPriceGraph<OpportunityMaint, CROpportunity>
{
  protected override DocumentMapping GetDocumentMapping()
  {
    return new DocumentMapping(typeof(CROpportunity))
    {
      BranchID = typeof(CROpportunity.branchID),
      BAccountID = typeof(CROpportunity.bAccountID)
      CuryInfoID = typeof(CROpportunity.curyInfoID)
     };
  }
  protected override DetailMapping GetDetailMapping()
  {
    return new DetailMapping(typeof(CROpportunityProducts))
    { …  };
  }
}
```

Acumatica
The Cloud ERP

Reusable Business Objects - Solution

- Better 1 time to see that 100 to hear:
- Examples:
  - ContactAddress
  - Discount
  - MultiCurrency
  - SalesPrice
  - SalesTax



Products Wholesale ▾ Source Code ☆

SCREEN ASPX | BUSINESS LOGIC | DATA ACCESS | FIND IN FILES | WEBSITE SOURCES

Find Text: SalesPriceGraph [FIND]

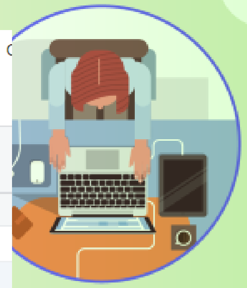
Name	Line	Content
PX.Objects\AR\ARInvoiceEntry.cs	5459	//public class SalesPrice : SalesPriceGraph<ARInvoiceEntry, ARInvoice>
PX.Objects\CR\OpportunityMaint.cs	2789	public class SalesPrice : SalesPriceGraph<OpportunityMaint, CROpportunity>
PX.Objects\CR\QuoteMaint.cs	1575	public class SalesPrice : SalesPriceGraph<QuoteMaint, CRQuote>
PX.Objects\Extensions\SalesPrice\SalesPriceGraph.cs	18	public abstract class SalesPriceGraph<TGraph, TPrimary> : PXGraphExtension<TGraph>

```
public abstract class SalesPriceGraph<TGraph, TPrimary> : PXGraphExtension<TGraph>
        where TGraph : PXGraph
        where TPrimary : class, IBqlTable, new()
{

    /// <summary>A class that defines the default mapping of the <see cref="Document" /> mapped cache extension to a DAC.</summary>
    protected class DocumentMapping : IBqlMapping
    {
        /// <exclude />
        protected Type _extension = typeof(Document);
        /// <exclude />
        public Type Extension => _extension;

        /// <exclude />
        protected Type _table;
        /// <exclude />
        public Type Table => _table;

        /// <summary>Creates the default mapping of the <see cref="Document" /> mapped cache extension to the specified table.</summary>
        /// <param name="table">A DAC.</param>
        public DocumentMapping(Type table)
```



Acumatica
The Cloud ERP

Generic Event Declaration

New Events Declaration is Available:

```
protected virtual void _(Events.FieldUpdated<DAC, DAC.field> e) {}
protected virtual void _(Events.FieldDefaulting<DAC, DAC.field> e) {}
protected virtual void _(Events.RowUpdated<DAC> e) {}
protected virtual void _(Events.RowSelected<DAC> e) {}
protected void FinPeriod(Events.CacheAttached<Batch.finPeriodID> e) {}
```

With new events no more mistakes in names

And you don’t need to cast DACs anymore

```
PXCache cache = e.Cache;
DAC row = e.Row;
```

Acumatica
The Cloud ERP

Generic Event Declaration

Notes:

- Both new and old events can (but shouldn't) be used simultaneously.
- Events are collected from the class in the declaration sequence.
  - New *ed events will be fired in original order (ascending)
  - New *ing events will be fired in reverse order (descending)

Limitations:

- CacheAttached events should have different names
- Override event approach with 2 parameters isn't yet supported

```
protected void _(Events.RowInserting<Batch> e, ~~PXRowInserting baseHandler~~)
```

Acumatica
The Cloud ERP

Further Presentations, More Patterns

- Code Patterns
  - Extract common logic from Graph and reuse it with Attributes and DataView
  - Usage of Events and Data References to pass additional information.
  - Dynamically subscribe for events
  - Pass Type parameters into attributes constructor to eliminate DAC dependency
  - IBqlUnary predicates
  - Declarative Referential Integrity Check
  - Signal R subscription from JavaScript
- Architecture Patterns
  - Virtual action declaration with Automation steps and pass parameters to code.
  - Design for records postprocessing
  - Processing of multiple records, Parallel Processing
- Unit Tests (including Reusable Business Objects)
- Logging in Acumatica.

Acumatica
The Cloud ERP

SUMMARY

Acumatica Developer Network

Acumatica Platform -Summary

Business Application

Platform

Technology / Environment / Framework / User Interface

- **Code Standardization –** it is easier to support standardized code by any developer who knows it
- **Speedup Development –** with high level primitives and tools
- **Protect form Technology changes –** platform hides underlaying technology and allows to change it without touching business logic code.
- **Share Tools –** involve all other companies to build new world together

Acumatica
The Cloud ERP

Thank You!

https://adn.Acumatica.com
http://asiablog.acumatica.com
https://github.com/smarenich/BlackBeltDevelopment