Black-Belt Developer Techniques

Part 1

Gabriel Michaud
Owner and Independent ERP Consultant
Velixo
gabriel@Velixo.com | @gabrielmichaud | blog.Velixo.com

Acumatica
DEVELOPER
MOST VALUABLE PROFESSIONAL

Welcome!

- Part 1: Everyday tips and tricks
- Part 2: Pushing the limits of Acumatica extensibility
- Part 3: Showcase and dev community call to action

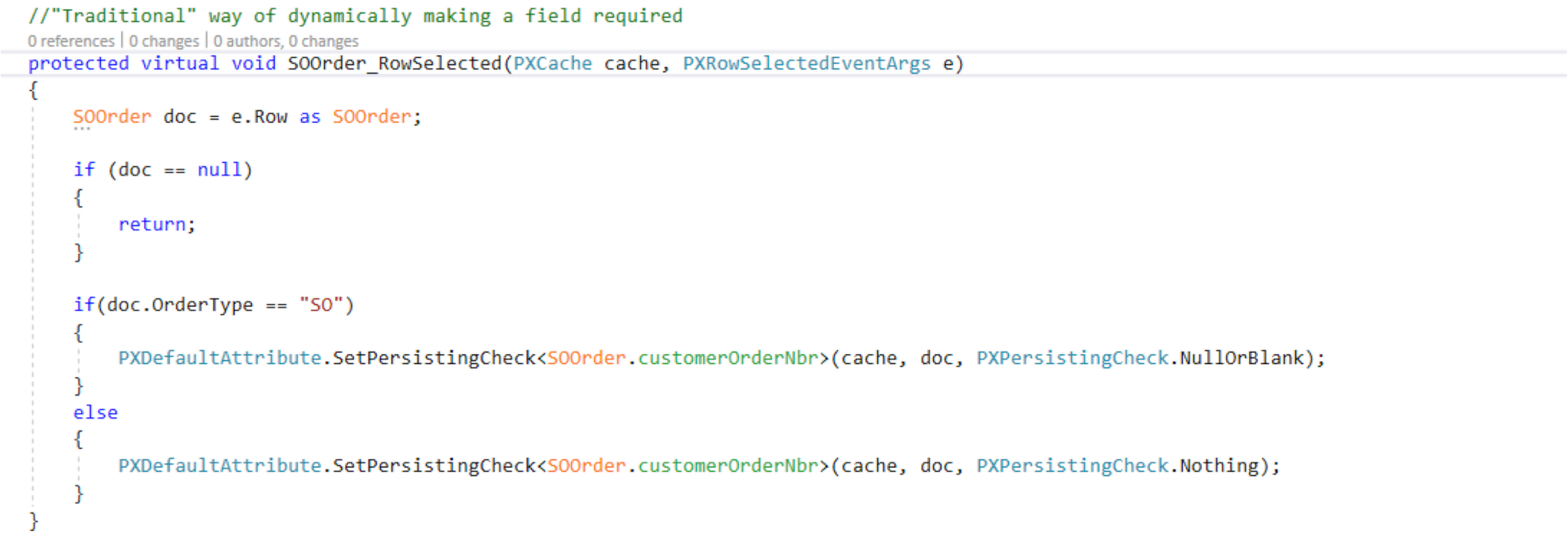
Part 1 – Everyday tips and tricks

What they don’t teach you at the (Acumatica) university

“Magic” Attributes and BQL Operands

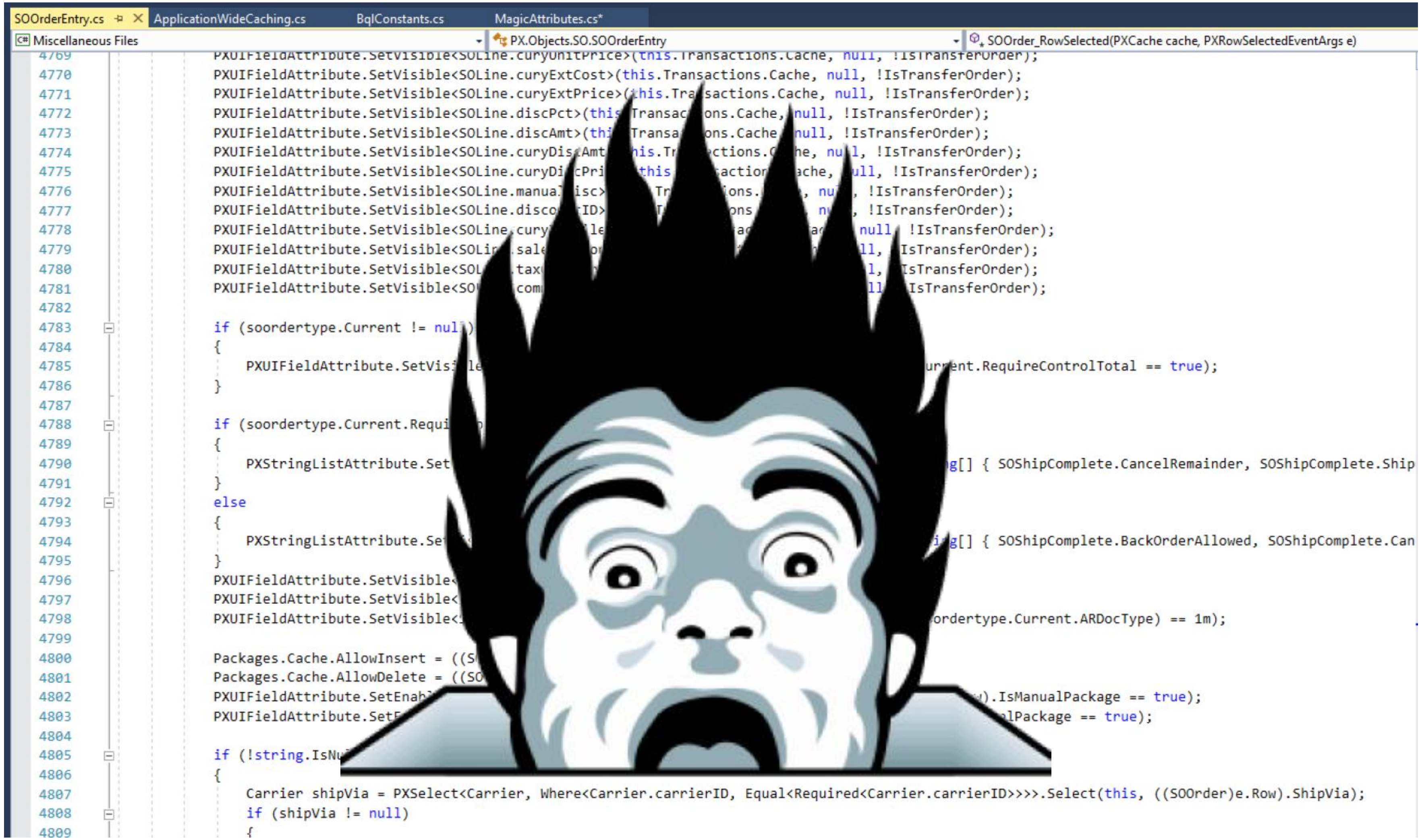
”The best code is no code at all” – Jeff Atwood (StackOverflow, Discourse)

The ”traditional” way of making a field required (or visible, or enabled) dynamically



```
//"Traditional" way of dynamically making a field required
0 references | 0 changes | 0 authors, 0 changes
protected virtual void SOOrder_RowSelected(PXCache cache, PXRowSelectedEventArgs e)
{
    SOOrder doc = e.Row as SOOrder;

    if (doc == null)
    {
        return;
    }

    if(doc.OrderType == "SO")
    {
        PXDefaultAttribute.SetPersistingCheck<SOOrder.customerOrderNbr>(cache, doc, PXPersistingCheck.NullOrBlank);
    }
    else
    {
        PXDefaultAttribute.SetPersistingCheck<SOOrder.customerOrderNbr>(cache, doc, PXPersistingCheck.Nothing);
    }
}
```

3 days (or hours) later…


SOOrderEntry.cs
ApplicationWideCaching.cs
BqlConstants.cs
MagicAttributes.cs*
Miscellaneous Files
PX.Objects.SO.SOOrderEntry
SOOrder_RowSelected(PXCache cache, PXRowSelectedEventArgs e)

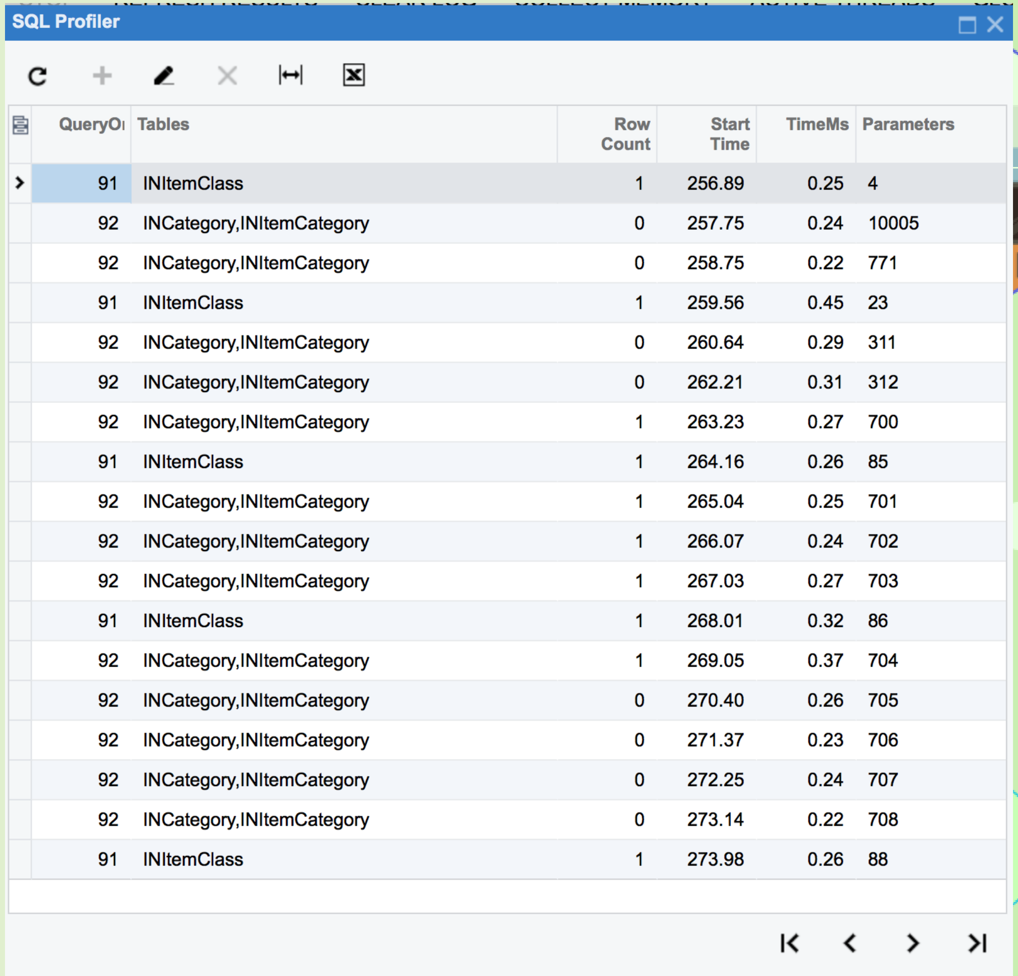
```
            PXUIFieldAttribute.SetVisible<SOLine.curyUnitPrice>(this.Transactions.Cache, null, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.curyExtCost>(this.Transactions.Cache, null, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.curyExtPrice>(this.Tra sactions.Cache, null, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.discPct>(this Transac ons.Cache, null, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.discAmt>(thi Transa ons.Cache null, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.curyDis Amt his.Tr ctions. he, nu l, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.curyDi cPri this action ache, ull, !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.manua isc> Tr ons. , nu , !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine.disco ID> T ons , nu , !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLine cury le a a null !IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOLi .sale o ll, IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SOL .tax l, IsTransferOrder);
            PXUIFieldAttribute.SetVisible<SO com ll IsTransferOrder);

            if (soordertype.Current != nul )
            {
                PXUIFieldAttribute.SetVis le                                              ur ent.RequireControlTotal == true);
            }

            if (soordertype.Current.Requi
            {
                PXStringListAttribute.Set                                                 g[] { SOShipComplete.CancelRemainder, SOShipComplete.Ship
            }
            else
            {
                PXStringListAttribute.Se                                                  g[] { SOShipComplete.BackOrderAllowed, SOShipComplete.Can
            }
            PXUIFieldAttribute.SetVisible<
            PXUIFieldAttribute.SetVisible<
            PXUIFieldAttribute.SetVisible<                                               ordertype.Current.ARDocType) == 1m);

            Packages.Cache.AllowInsert = ((S
            Packages.Cache.AllowDelete = ((SO
            PXUIFieldAttribute.SetEnab                                                   ).IsManualPackage == true);
            PXUIFieldAttribute.SetE                                                      lPackage == true);

            if (!string.IsNu
            {
                Carrier shipVia = PXSelect<Carrier, Where<Carrier.carrierID, Equal<Required<Carrier.carrierID>>>>.Select(this, ((SOOrder)e.Row).ShipVia);
                if (shipVia != null)
                {
```

“Magic” Attributes and BQL Operands

- PXUIRequired(Type condition)
- PXUIEnabled(Type condition)
- PXUIVisible(Type condition) -> New in 2018 R1 Update 4
- PXUIVerify(Type condition)
- Related Operands
  - Selector<KeyField, Operand>
  - Default<Field, …>

Application-wide data caching with slots and IPrefetchable

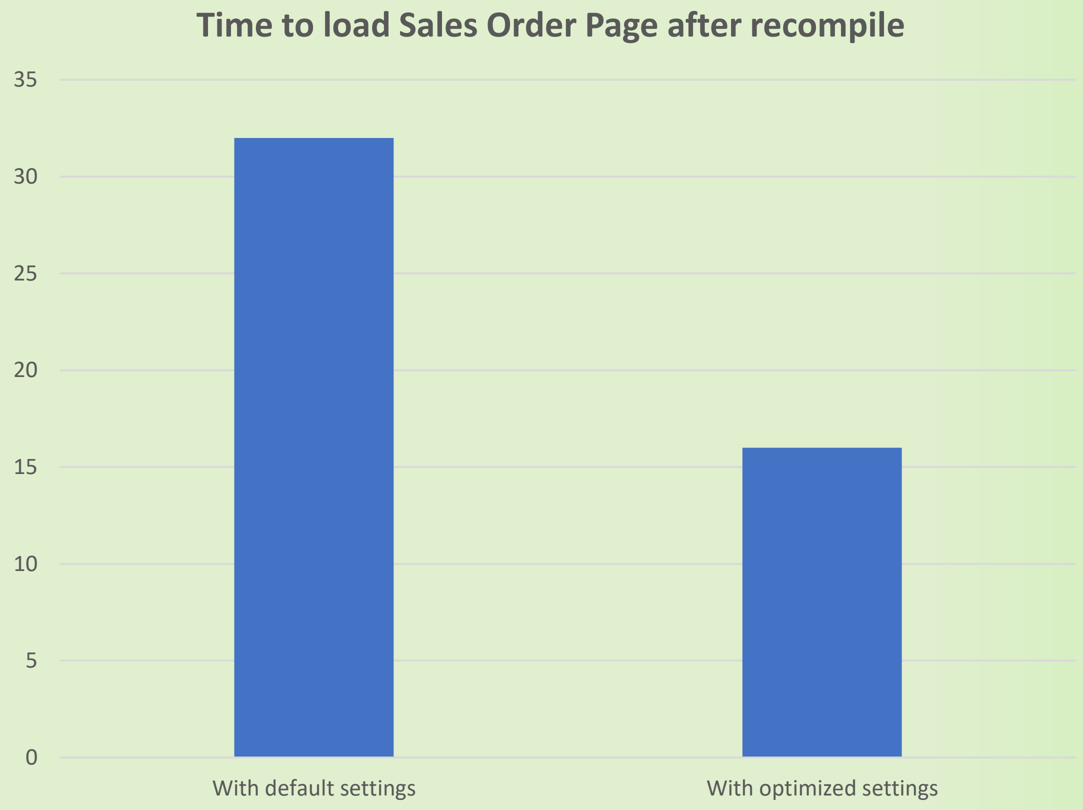
- Use case: frequently accessed data that is expensive to load, reasonable in size and not updated frequently
- Existing examples within Acumatica: user permissions, site map, segmented key configuration, discount rules
- Acumatica takes care of initializing it on first access and invalidating the (whole) cache when a row changes



	QueryO	Tables	Row Count	Start Time	TimeMs	Parameters
›	91	INItemClass	1	256.89	0.25	4
	92	INCategory,INItemCategory	0	257.75	0.24	10005
	92	INCategory,INItemCategory	0	258.75	0.22	771
	91	INItemClass	1	259.56	0.45	23
	92	INCategory,INItemCategory	0	260.64	0.29	311
	92	INCategory,INItemCategory	0	262.21	0.31	312
	92	INCategory,INItemCategory	1	263.23	0.27	700
	91	INItemClass	1	264.16	0.26	85
	92	INCategory,INItemCategory	1	265.04	0.25	701
	92	INCategory,INItemCategory	1	266.07	0.24	702
	92	INCategory,INItemCategory	1	267.03	0.27	703
	91	INItemClass	1	268.01	0.32	86
	92	INCategory,INItemCategory	1	269.05	0.37	704
	92	INCategory,INItemCategory	0	270.40	0.26	705
	92	INCategory,INItemCategory	0	271.37	0.23	706
	92	INCategory,INItemCategory	0	272.25	0.24	707
	92	INCategory,INItemCategory	0	273.14	0.22	708
	91	INItemClass	1	273.98	0.26	88

Optimizing your development environment – web.config tweaks

- `<appSettings>`
  - CompilePages = FALSE
  - InstantiateAllCaches = FALSE
- `<compilation>`
  - OptimizeCompilations = TRUE



Part 2 - Pushing the limits of Acumatica extensibility

Unsupported techniques that can save the day or land you in trouble

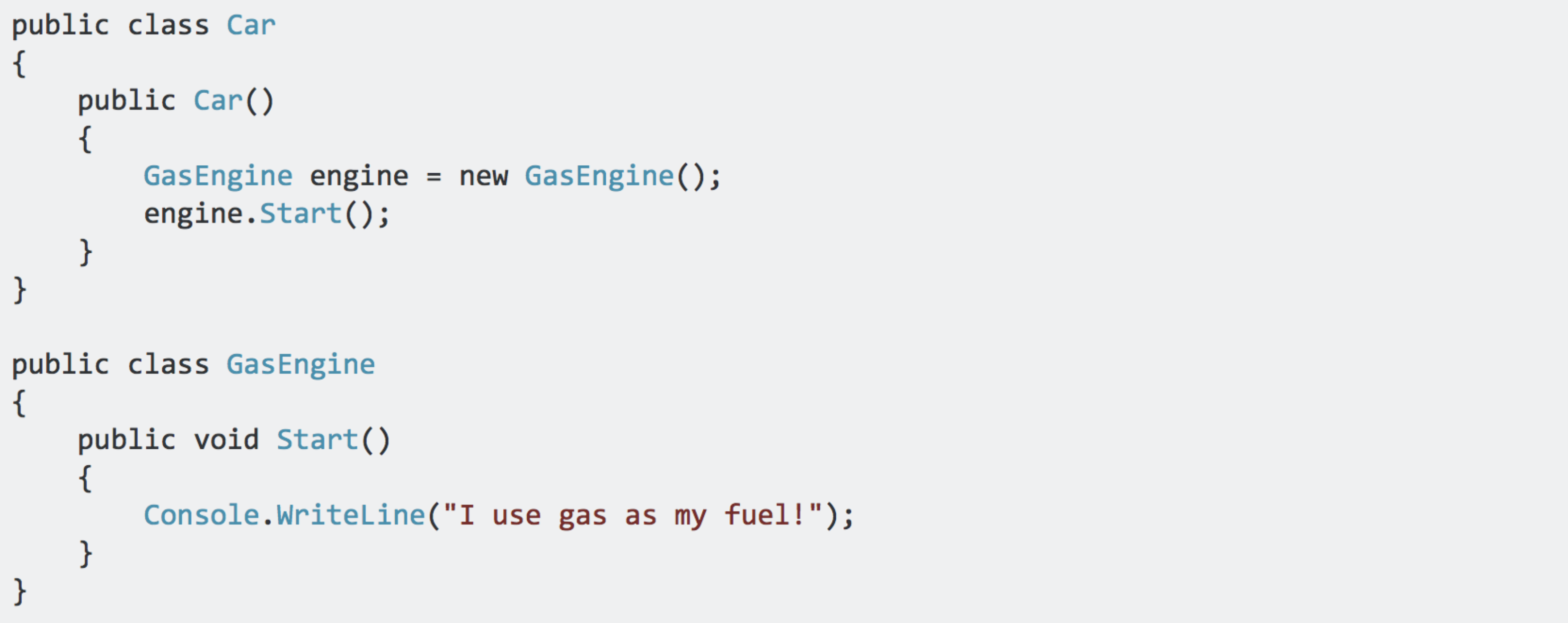
DANGER
HARD HAT
AREA
©2010JJP

Über-graph extensions

- PXGraphExtension<Graph> where Graph : PXGraph
- Ex: SOEntryExt : PXGraphExtension<SOOrderEntry>
- What would happen if we did PXGraphExtension<PXGraph> ?
- Demos
  - Adding an action to every graph
  - Hooking into an event (Note field updated)

Dependency Injection with Acumatica

- Traditionally, each object is responsible for obtaining its own references to the dependent objects it collaborates with:

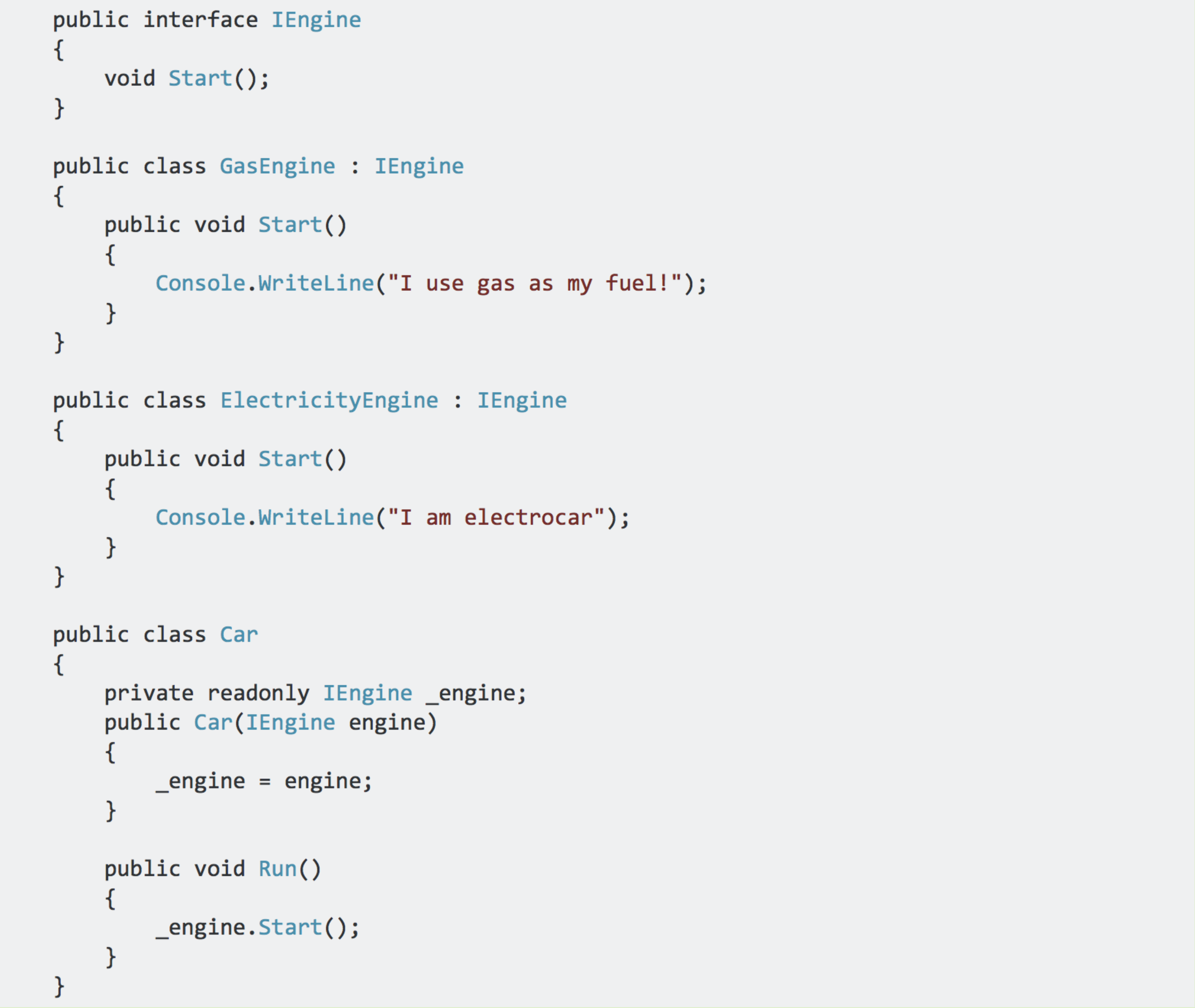


```
public class Car
{
    public Car()
    {
        GasEngine engine = new GasEngine();
        engine.Start();
    }
}

public class GasEngine
{
    public void Start()
    {
        Console.WriteLine("I use gas as my fuel!");
    }
}
```

- Problems: Tightly coupled, hard to test

Source: https://stackoverflow.com/a/44945310/624448

Dependency Injection with Acumatica

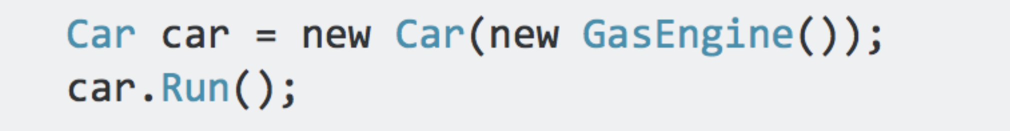


```
public interface IEngine
{
    void Start();
}

public class GasEngine : IEngine
{
    public void Start()
    {
        Console.WriteLine("I use gas as my fuel!");
    }
}

public class ElectricityEngine : IEngine
{
    public void Start()
    {
        Console.WriteLine("I am electrocar");
    }
}

public class Car
{
    private readonly IEngine _engine;
    public Car(IEngine engine)
    {
        _engine = engine;
    }

    public void Run()
    {
        _engine.Start();
    }
}
```

Source: https://stackoverflow.com/a/44945310/624448

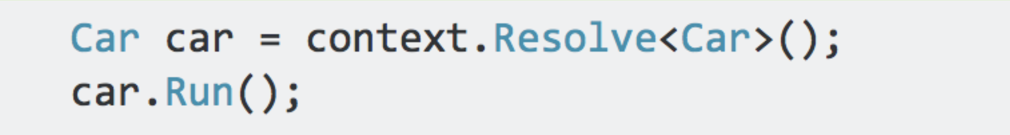
Dependency Injection with Acumatica

- Getting a new Car instance, the old fashioned way:



```
Car car = new Car(new GasEngine());
car.Run();
```

- With the help of Dependency Injection:



```
Car car = context.Resolve<Car>();
car.Run();
```

Dependency Injection with Acumatica

- **Summary**: DI is a technique for achieving loose coupling between objects and their dependencies. Rather than directly instantiating dependencies that class needs in order to perform its actions, dependencies are provided to the class by someone else
- We typically delegate that responsibility to a framework:
  - Structure Map
  - Unity
  - Ninject
  - Castle Windsor
  - Autofac -> used by Acumatica

autofac

- Current (public) uses inside Acumatica
  - Generic graph extensions (ref: 2018 R1 Framework Development Guide)
- Demo

Custom HTTP Handlers

- Made possible by the Dependency Injection framework and ActivateOnApplicationStart
- Demo

WITH

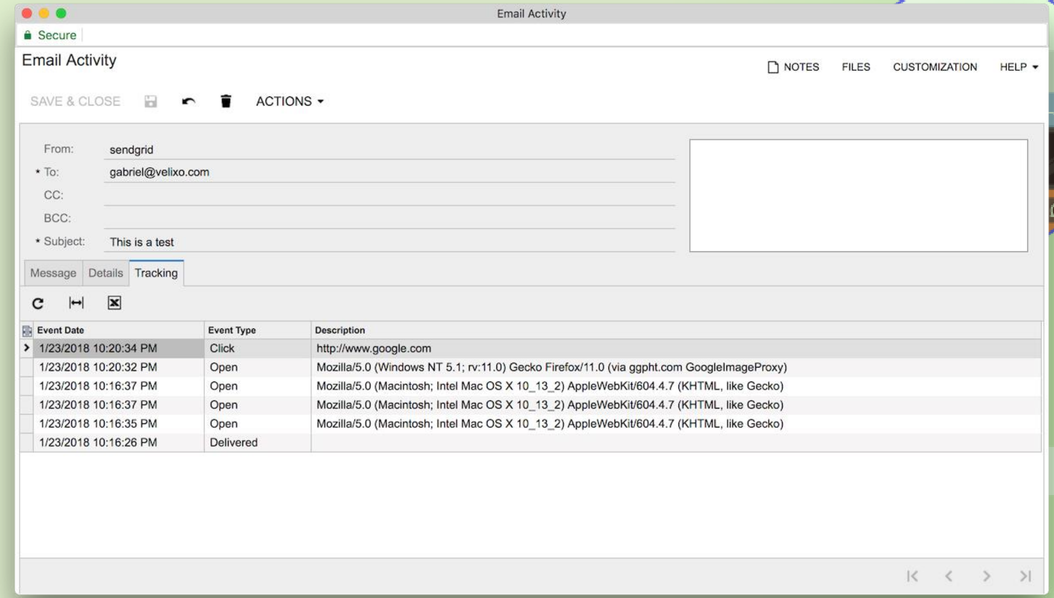
GREAT POWER COMES

GREAT RESPONSIBILITY

Part 3 – Showcase and dev community call to action

Velixo Mail Track: Acumatica/SendGrid Integration

- Tracks delivery, open and click-through
- Leverages Chrome push notifications to notify users instantly when an event occurs
- Techniques: custom HTTP module, Chrome push notifications, source control automatic command-line build
- Status: Production-ready
- Link: github.com/gmichaud/Velixo-MailTrack



Email Activity

From: sendgrid

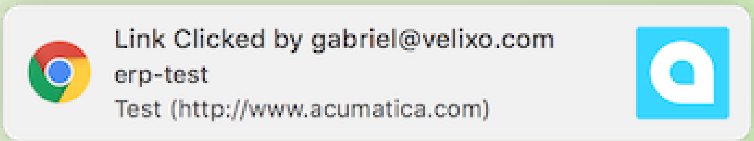
* To: gabriel@velixo.com

CC:

BCC:

* Subject: This is a test

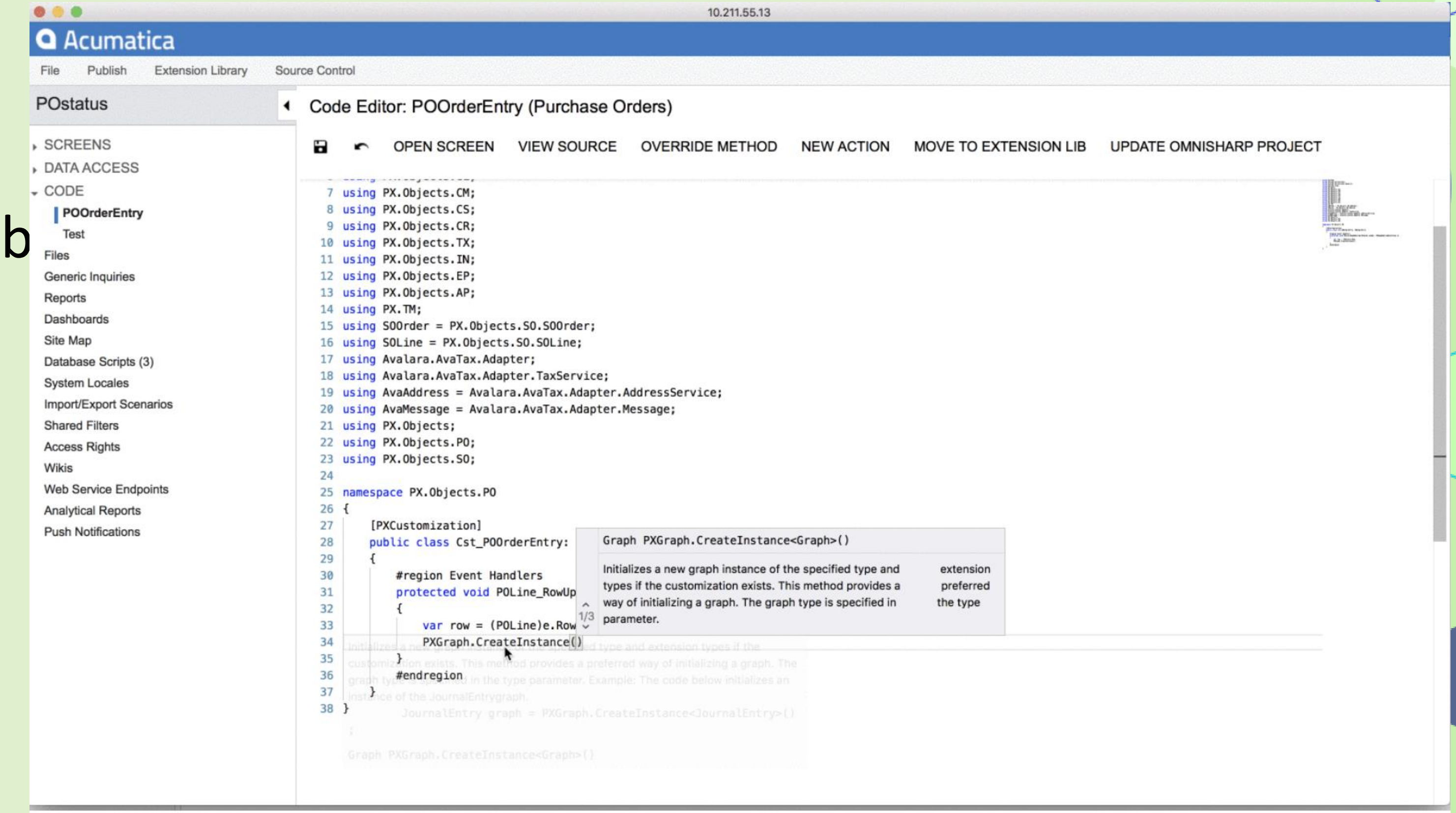
Event Date	Event Type	Description
1/23/2018 10:20:34 PM	Click	http://www.google.com
1/23/2018 10:20:32 PM	Open	Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko Firefox/11.0 (via ggpht.com GoogleImageProxy)
1/23/2018 10:16:37 PM	Open	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_2) AppleWebKit/604.4.7 (KHTML, like Gecko)
1/23/2018 10:16:37 PM	Open	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_2) AppleWebKit/604.4.7 (KHTML, like Gecko)
1/23/2018 10:16:35 PM	Open	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_2) AppleWebKit/604.4.7 (KHTML, like Gecko)
1/23/2018 10:16:26 PM	Delivered	



Link Clicked by gabriel@velixo.com
erp-test
Test (http://www.acumatica.com)

Velixo Code Editor for Acumatica

- Techniques: custom HTTP module, custom title module, global graph extensions,PXDatabase.Subscribe<>, invoking private functions through reflection, embedding and loading external EXE file, replacing system pages
- Status: Experimental
- Link: github.com/gmichaud/Velixo-AcumaticaCodeEditor



Code Editor: POOrderEntry (Purchase Orders)

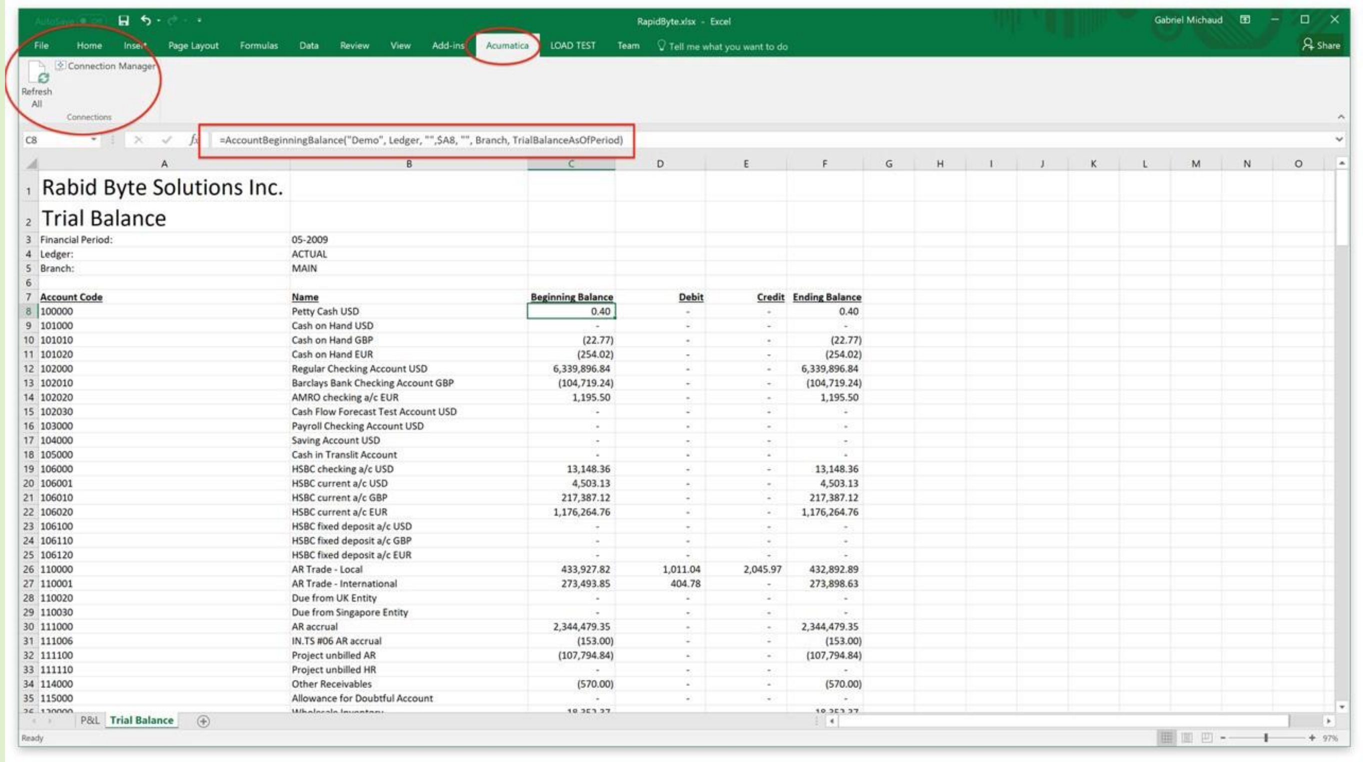
```
using PX.Objects.CM;
using PX.Objects.CS;
using PX.Objects.CR;
using PX.Objects.TX;
using PX.Objects.IN;
using PX.Objects.EP;
using PX.Objects.AP;
using PX.TM;
using SOOrder = PX.Objects.SO.SOOrder;
using SOLine = PX.Objects.SO.SOLine;
using Avalara.AvaTax.Adapter;
using Avalara.AvaTax.Adapter.TaxService;
using AvaAddress = Avalara.AvaTax.Adapter.AddressService;
using AvaMessage = Avalara.AvaTax.Adapter.Message;
using PX.Objects;
using PX.Objects.PO;
using PX.Objects.SO;

namespace PX.Objects.PO
{
    [PXCustomization]
    public class Cst_POOrderEntry:
    {
        #region Event Handlers
        protected void POLine_RowUp
        {
            var row = (POLine)e.Row
            PXGraph.CreateInstance()
        }
        #endregion
    }
}
```

```
Graph PXGraph.CreateInstance<Graph>()
```

Initializes a new graph instance of the specified type and extension types if the customization exists. This method provides a preferred way of initializing a graph. The graph type is specified in the type parameter.

Velixo Reports: Analytical Reporting in Excel

- The ease of use of Excel with the power of Acumatica Analytical Reports
- Status: Production-ready (commercial version available)
- Link: github.com/gmichaud/Velixo-Reports
- More on my blog: blog.velixo.com



=AccountBeginningBalance("Demo", Ledger, "",$A8, "", Branch, TrialBalanceAsOfPeriod)

Rabid Byte Solutions Inc.

Trial Balance

Financial Period:	05-2009
Ledger:	ACTUAL
Branch:	MAIN

Account Code	Name	Beginning Balance	Debit	Credit	Ending Balance
100000	Petty Cash USD	0.40	-	-	0.40
101000	Cash on Hand USD	-	-	-	-
101010	Cash on Hand GBP	(22.77)	-	-	(22.77)
101020	Cash on Hand EUR	(254.02)	-	-	(254.02)
102000	Regular Checking Account USD	6,339,896.84	-	-	6,339,896.84
102010	Barclays Bank Checking Account GBP	(104,719.24)	-	-	(104,719.24)
102020	AMRO checking a/c EUR	1,195.50	-	-	1,195.50
102030	Cash Flow Forecast Test Account USD	-	-	-	-
103000	Payroll Checking Account USD	-	-	-	-
104000	Saving Account USD	-	-	-	-
105000	Cash in Translit Account	-	-	-	-
106000	HSBC checking a/c USD	13,148.36	-	-	13,148.36
106001	HSBC current a/c USD	4,503.13	-	-	4,503.13
106010	HSBC current a/c GBP	217,387.12	-	-	217,387.12
106020	HSBC current a/c EUR	1,176,264.76	-	-	1,176,264.76
106100	HSBC fixed deposit a/c USD	-	-	-	-
106110	HSBC fixed deposit a/c GBP	-	-	-	-
106120	HSBC fixed deposit a/c EUR	-	-	-	-
110000	AR Trade - Local	433,927.82	1,011.04	2,045.97	432,892.89
110001	AR Trade - International	273,493.85	404.78	-	273,898.63
110020	Due from UK Entity	-	-	-	-
110030	Due from Singapore Entity	-	-	-	-
111000	AR accrual	2,344,479.35	-	-	2,344,479.35
111006	IN.TS #06 AR accrual	(153.00)	-	-	(153.00)
111100	Project unbilled AR	(107,794.84)	-	-	(107,794.84)
111110	Project unbilled HR	-	-	-	-
114000	Other Receivables	(570.00)	-	-	(570.00)
115000	Allowance for Doubtful Account	-	-	-	-

Thank You!

Code samples and presentation: https://github.com/gmichaud/BlackBeltTechniques

https://adn.Acumatica.com