CS3 Technology

Customize ERP

Prepared by: Troy Vars
Date Prepared: 12/10/18

Executive Summary: Choosing the best fit ERP for you should entail criteria for your anticipated customization level. As with any endeavor involving an ERP system, customization should have a strong Business Case to proceed. Achieving your Return on Investment (ROI) is a key indicator to a successful project implementation. One should never change just to change. Customizations come with a hidden cost in maintenance that should be offset by the efficiency and process gains from your Business Case. Depending on the software manufacturer's openness to customizations, the software/technology stacks age, and customizers resource availability the costs of such customization can range from miniscule to extravagant.

These costs originate from additional testing and fixes before applying upgrades from a software manufacturer, continued development as your business grows and evolves, and expensive redesigns and reworking due to constant technology evolution. Not all of which are under a business's control.

To Customize, or Not to Customize?

So out of the Box is Better? In most cases, yes, but not necessarily. Depending on your industry, ERP choices, and individual processes some customizing might be the right fit for you, but it is highly unlikely that your company's situation is so unique that a scratch build is needed.

Often in the software industry we hear that a company's processes are unique to their industry. In practice, I find that tailoring a system to a process is beneficial roughly half the time. In many cases a company's processes are a matter of tradition more so than process engineering. We look at our old system or process under the cry of "We/it used to do X!" from a very human fear to change. A few simple questions can often make the necessity to customize, or not, self-evident.

- Why & When did we start doing it?
    - Was it in response to a certain event?
        - Customer upset?
        - Revenue loss? Cost overruns?
        - Temporary metrics?
    - Did it arise because a previous system or process required it?
        - Work arounds?
        - Broken process?
        - Band-aid solution?
- What financial gain is there to doing it?
    - Does it generate revenue directly?
    - Does it add value to the item/service your selling?
    - Does it provide insight into the process that cannot be gained by any other means?
    - Does it eliminate significant waste further down in the process?
- If we didn't do it what would be the cost/impact?
    - Would revenue be lost?
    - Would time/cost/quality be decreased?
- Why doesn't benchmark or similar Industry z do it if we do?

If your Subject Matter Experts (SME) and managers are looking around the room at each other or answering, “Just Because we always have.” You probably want to look at a process change before making a system change.

Types of Customizations

Layout Changes are the simple changes. Swapping the field order on a form for ease of data entry. Manipulating layouts of menus. Renaming fields for better end user understanding or to fit to a company’s corporate glossary. Adding new fields or sections for data entry. These tend to be lower end cost.

Workflow Changes are middle ground area. They entail hiding or unhiding fields depending on where in the process a particular record is. Blocking an employee from modifying the picked qty after the picking step is complete or preventing the reopening of a case based on status.

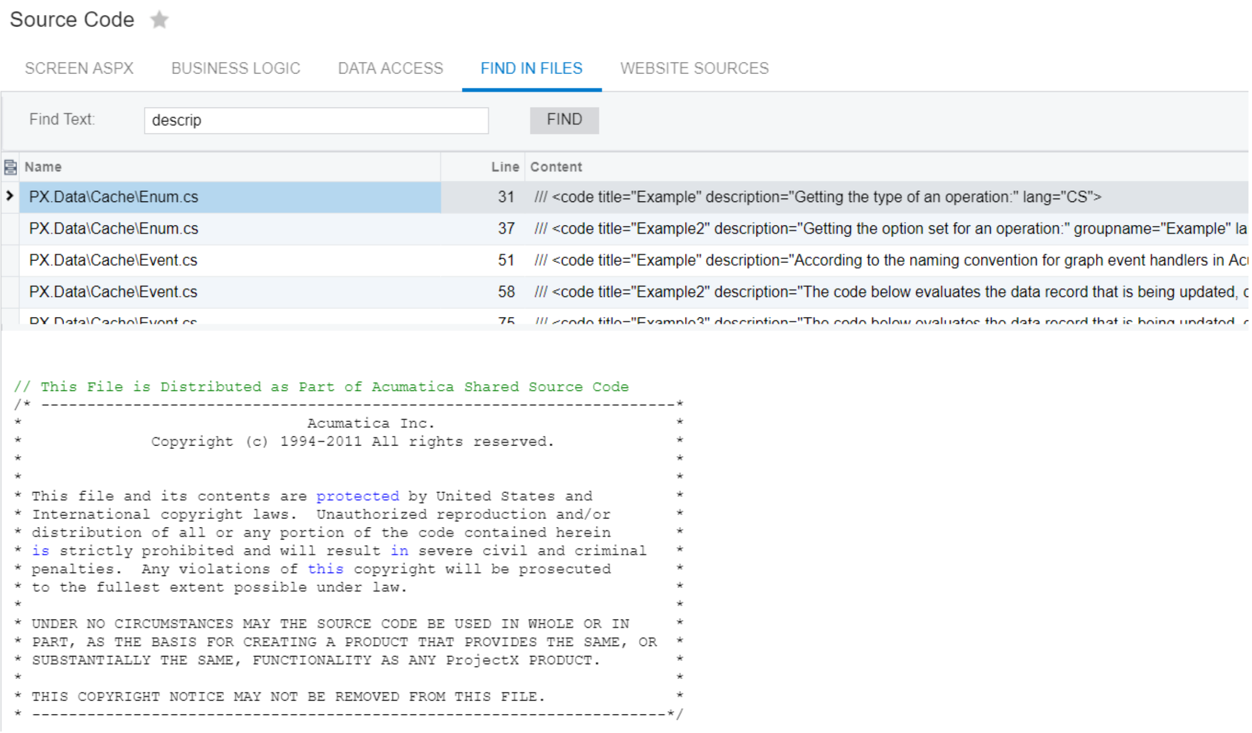
Business Logic Changes are the most complete end user changes. The validating of one field based on multiple other fields. Actions such as copying new fields to further records or starting a new form based on the end user’s inputs on this form. Creating a custom calculation for availability that ignores stock purchased for specific projects. Any system already has a built-in logic by which operates. Modifying that behavior requires knowing what else is dependent on that logic. Much like removing a step from a business process requires understanding what you are losing or gaining, customizing business logic requires a clear idea of what is dependent on that step.

Cross-system Integrations are data, api or interface sync points that allow data signals to flow from one system to another system. Much like emails or paperwork that flows between desks in an office, these signals trigger the separate systems to begin new work or to make the contained information available to the end user. This allows outside functionality to be used from the ERP system data. Expanding the usefulness of the ERP solution without costly development to move the outside functionality into ERP, nor forcing the end users to duplicate enter into multiple systems.

Acumatica and Customization

When looking at Acumatica, I personally have found that the company is committed to enabling and supporting customization. They have embraced the fact that in our world one size does not fit all. All their decisions from the stack its built upon to the development of additional customization tools demonstrate their affability to the longevity and customization friendly ERP they developed.

Their chosen stack is .NET Framework with C# code base is a long supported and widely utilized Microsoft owned programming language. In addition, their use of ASP.NET supports Javascript and PHP, two of the most popular web development languages. This stack makes for a stable and minimally impacted customization environment. They make the majority of their source code available to developers to refer to and modify.

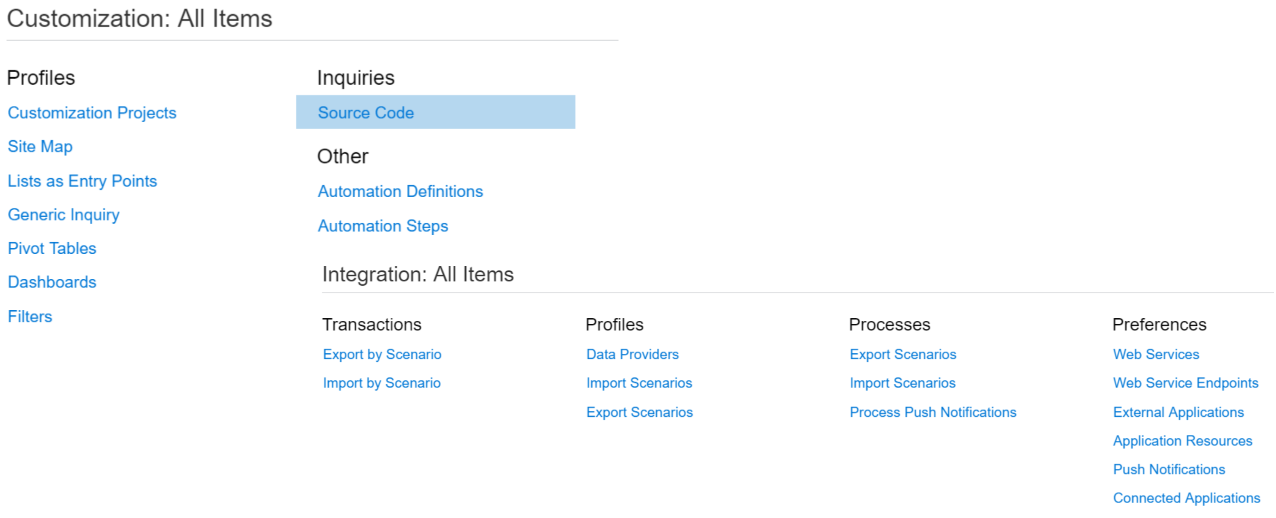


Name	Line	Content
PX.Data\Cache\Enum.cs	31	/// <code title="Example" description="Getting the type of an operation:" lang="CS">
PX.Data\Cache\Enum.cs	37	/// <code title="Example2" description="Getting the option set for an operation:" groupname="Example" la
PX.Data\Cache\Event.cs	51	/// <code title="Example" description="According to the naming convention for graph event handlers in Ac
PX.Data\Cache\Event.cs	58	/// <code title="Example2" description="The code below evaluates the data record that is being updated, c
PX.Data\Cache\Event.cs	75	/// <code title="Example3" description="The code below evaluates the data record that is being updated, c

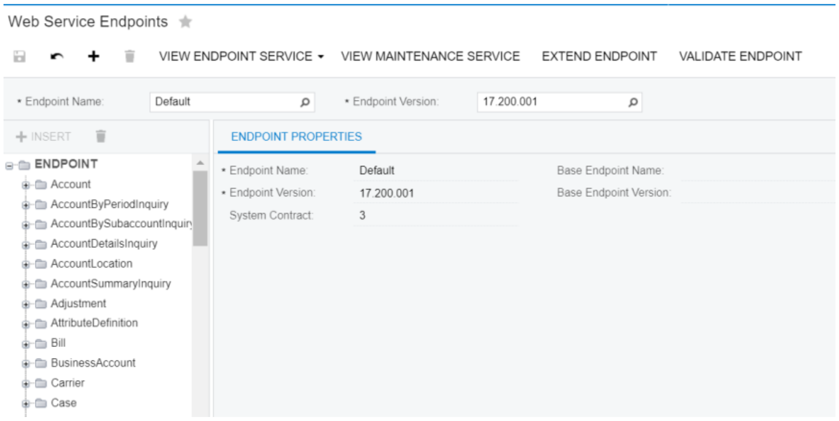
```
// This File is Distributed as Part of Acumatica Shared Source Code
/* ---------------------------------------------------------------------*
*                               Acumatica Inc.                          *
*              Copyright (c) 1994-2011 All rights reserved.             *
*                                                                       *
*                                                                       *
* This file and its contents are protected by United States and        *
* International copyright laws.  Unauthorized reproduction and/or      *
* distribution of all or any portion of the code contained herein     *
* is strictly prohibited and will result in severe civil and criminal  *
* penalties.  Any violations of this copyright will be prosecuted      *
* to the fullest extent possible under law.                            *
*                                                                       *
* UNDER NO CIRCUMSTANCES MAY THE SOURCE CODE BE USED IN WHOLE OR IN    *
* PART, AS THE BASIS FOR CREATING A PRODUCT THAT PROVIDES THE SAME, OR  *
* SUBSTANTIALLY THE SAME, FUNCTIONALITY AS ANY ProjectX PRODUCT.       *
*                                                                       *
* THIS COPYRIGHT NOTICE MAY NOT BE REMOVED FROM THIS FILE.             *
* ---------------------------------------------------------------------*/
```

Acumatica’s Roadmap includes two major yearly releases meaning that if a company update bi-annually or annually they will always be on the cutting edge of efficiency and benefit from the development budget of the software manufacturer. The underlying table structure of Acumatica uses a naming convention so that even if they come up with a similar named field or process, you don’t have to worry about data loss to your custom added fields.

Additionally, several powerful tools are available within the application that promote ease of customization. The customization tool allows C# editing inside of Acumatica as well providing user friendly tools for the basic layout, workflow and business logic customizations.



The newest tools support data and api integrations from user friendly screens that further streamline the process of tying disparate systems and data sources together. Every screen in Acumatica is available to SOAP API via WSDL. Any SOAP actions are sent thru the same Business Logic as the end user inputs even if that logic has been customized. Ensuring you get the same data from your employee input as an integrated Web Form hosted on your website.

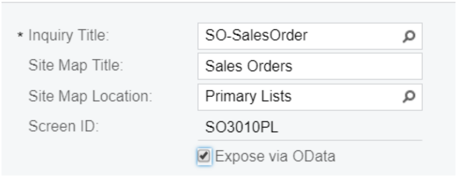


* Endpoint Name:	Default	* Endpoint Version:	17.200.001

* Endpoint Name:	Default	Base Endpoint Name:	
* Endpoint Version:	17.200.001	Base Endpoint Version:	
System Contract:	3		

Web Service End Points and Extended Applications opens opportunities to use Contract SOAP APIs to make it possible for web hosted external applications to be part of your Acumatica experience.

Making such tools as Twilio, SmartSheet, and Docusign available to end users from inside of Acumatica with limited integration time. Recently, they added the ability to send push notifications out via WebHooks, SignalR and MSMQ to programs like Microsoft Flow. This brings a much more robust set of integration options to the table. I excited to see what 2019 brings!



* Inquiry Title:	SO-SalesOrder
Site Map Title:	Sales Orders
Site Map Location:	Primary Lists
Screen ID:	SO3010PL

☑ Expose via OData

Lastly, a suite of report customization and data tools are packaged with Acumatica to make custom reports and Key Process Indicator (KPI) metrics available inside and outside of Acumatica. This allows integrations to Power BI and Tableau via Odata. Changes to the layout of forms and reports are handled thru the report designer. So many great tools, I could go on infinitum.

If you would like me to, consider a demo of Acumatica from CS3…

About the Author

Troy Vars is a Senior Consultant, Web Developer and 3-year Acumatica Convert. After being a 20-year veteran in the struggle to balance end user satisfaction and system upgradability in the SAP and Web Development world he has joined CS3.