Contents

Enriching the User Experience with Agile Business Suite

By Thangathen Ponnusamy, Global Product Manager – Agile Business Suite and Enterprise Application Environment, Unisys

User experience and user interface (UI) technologies open new doors for engaging your clients, partners, and employees.

All too often, though, connecting with these groups means balancing two important, yet competing, factors. On one hand, business processes – and the enterprise applications that encompass them – have become increasingly complex as they incorporate new initiatives and growing client demands. At the same time, the simplicity and usability of these applications frequently determines how successful they'll ultimately be.

Users demand aesthetically pleasing, minimalist design. Form and function must be tightly integrated, with only the most necessary, relevant information provided at all times. The UI should meet all pertinent functional requirements while making it easy for users to be productive – without rigorous, time-consuming training. It must be tailored to end users' unique knowledge, as well as the context in which they choose to work. And in most cases, it needs to get to market as fast as possible.

With Agile Business Suite (AB Suite®), you have the freedom to utilize a rich set of tools that generates the UI for you, helping you avoid time constraints, reduce your dependence on specialized skills, and get new capabilities to market faster.

Client Tools

With AB Suite Client Tools, a suite of software that provides a variety of UIs for AB Suite applications, you're able to generate a wide range of end-user interfaces for desktop, web, and mobile devices, as well as service-based applications. >>

Not a subscriber of Developing Agility? Don't miss the next issue – sign up today.
To see earlier issues of Developing Agility, please visit the archive.

Because Client Tools creates generated solutions that deliver predictable results – along with tested code, stability, performance, and scalability – you'll get to market faster. And by providing a rich user experience, Client Tools makes applications simpler to learn, easing training requirements for both EAE and AB Suite end users, while increasing their productivity and promoting greater flexibility.

You can use either the standard generators or develop a custom generator with the help of the software's Customization Kit. With the Customization Kit, you'll receive guidance for building a generator yourself, or selecting one provided by a third party. No matter which route you choose, you'll be able to create a UI that utilizes the definitions in your AB Suite model, reflects your requirements, and is built using the technology you prefer. And because Client Tools is common across EAE and AB Suite, modernizing your EAE interfaces – and, ultimately, making the transition to AB Suite – will become much easier because the UI will be the same.

ClearPath Forward ePortal

You can transform existing AB Suite application interfaces using the point-and-click modernization environment offered in the ClearPath Forward® ePortal. With ePortal Developer, you can generate the complete solution and then stage it for deployment, all while you stay focused on what's most important: optimally presenting the application to your end users. And because automation is used across the entire solution, you'll get your modernized application to market faster and more cost efficiently – without the need for specialized skills.

Pair these capabilities with the innovative new Unisys Mobile Hybrid Application Build Service, and you can further enrich your applications by combining them with the native capabilities of smart device services, such as GPS, barcode scanning, and photo and video capture.

Client Framework

With the Agile Business Suite Client Framework, you can develop state-of-the-art UIs using your preferred technology. You'll get total control over the user experience, making it easier to deliver more satisfying results to the individuals who interact with your applications every day. And you'll be able to separate UI components from the application's business logic and database. This way, your UI designers can focus on delivering rich user experiences, while your AB Suite developers work to align business logic and interface definitions with the latest requirements.

As a supported part of the AB Suite Runtime for Microsoft® Windows®, the Client Framework enables you to use the popular Windows Presentation Foundation (WPF) technology to create UIs based on the Data Models you generate. In addition, the Client Framework works with such tools as ASP.Net MVC 5, ASP.Net Web API, HTML5, JavaScript, CSS, and JQuery.

Enrich Your Client Experience

AB Suite offers a number of UI technologies you can easily utilize to enrich the user experience and more effectively engage external and internal clients alike. If you'd like to learn more about the capabilities described in this article, or wish to explore them further in a pilot project, we encourage you to contact us at ABSuite@unisys.com.

With the planned 2Q 2019 availability of Agile Business Suite Release 7.0 drawing nearer by the day, we've been encouraging the AB Suite user community to take part in a number of beta tests.

In fact, we recently completed the AB Suite 7.0 Beta 4 test. This is always an exciting test, as the Beta 4 software usually includes much of the functionality that will find its way into the final release. Here are some of the key new features users explored during the Beta 4 test.

AB Suite Modes

Each AB Suite model can be configured to use one of several Product Feature Modes. These unique operating modes, which expose specific functionality, have been expanding over the last couple AB Suite releases, and they continue to evolve in AB Suite 7.0.

Based on feedback we received following earlier beta tests, the naming conventions for the Product Feature Modes were updated ahead of the Beta 4 test. The updated mode names and the functionality each includes are now:

- **Standard:** Core object-oriented features. Available on the runtimes for ClearPath® MCP and Windows.
- **Advanced:** Advanced polymorphic features. Available on MCP and Windows, though XML is not supported on the MCP runtime.
- **User Experience:** Agile Business Suite Client Framework UI features. Available on Windows only.

AB Suite Model

The AB Suite Model redesigned the Class in the Beta 4 test to only represent complex types. As covered in a previous beta test, Primitive types are defined with the new Primitive object. Other notable changes covered during the Beta 4 test include:

- Extract files are defined using the new "File" stereotype, as opposed to inheriting from the GLB.File
- Attribute value-checking logic is available only on Ispec attributes in the presentation
- The model rules and validation regarding where inner classes can be instantiated have been improved, making it easier to reuse these classes

AB Suite Debugger

Beta 4 testers were able to use the new auto configure feature in Debugger, which populates the minimum configuration properties with default values when a debug session is initiated. Once testers started their debug sessions, they were able to modify the default properties in the model's "Configuration Properties" dialog box.

Testers could also try updated "Watch," "Local," and "Auto" windows, all of which have been enhanced with full edit capabilities, as well as the ability to monitor individual cells in an array. The display of large arrays and groups as a single string was moved to the new "Immediate" window, making it an on-demand display. Testers noted this significantly improved performance. >>

AB Suite Debugger and ATT

Testers were able to play back recorded tests against Debugger for "user experience" project types.

AB Suite Command-Line Builds

As of the Beta 4 test, the command-line utility "Builder.exe" has been retired. All builds from the command line are now made via MS Build. In addition, testers could also build and deploy systems for Debugger using MS Build. To enable this, the Debugger user settings are now stored in a ".user" file in the project directory.

AB Suite on Windows

The Beta 4 test included side-by-side support for the AB Suite on Windows runtime. This allowed users to install release 7.0 alongside other versions of AB Suite on the same machine without any conflicts.

Business Integrator

Because the model files generated by Business Integrator are now compatible with the AB Suite Product Feature Modes, testers could load the respective model files required by each mode.

User Documentation

The Information Center introduced in the Beta 4 test gave users a single, searchable, central access point to all of the release's key documentation.

As always, we greatly appreciate our beta testers taking the time to put AB Suite 7.0 through its paces. If you'd like to take part in future beta tests, please reach out to us at ABSuite@unisys.com.

Engineering Corner:
Calling Web Services Using WebAppSupport

By Howard Bell, Architect — EAE/AB Suite MCP Runtime and Debugger, Unisys

With the new Web Service Wizard in Agile Business Suite 6.1, you can easily invoke a Web Service in your ClearPath MCP application that encapsulates the functionality used by your Reports and Ispecs.

Using the wizard is simple. To create an MCP Web Service, perform the following steps:

1. Right-click the segment, then select Add>Add New Item.

2. In the "Add New Item" dialog box, select Web Service>Create. The Web Service Wizard will be displayed.

3. Enter a name for your Web Service class, then click "Finish." The example below keeps the default Web Service class name, "WebService1."

The created class contains the pre-loaded logic and attributes required to exercise the Web Service.

Here's how it looks when used in conjunction with the Unisys Weather Service. You can invoke the method from an Ispec or Report by setting the WeatherStnCode to any valid ICAO code to test it. For example, choosing "PHNL" will allow you to view the weather in Hawaii.

```
1    :Invoke the Weather Service using a valid ICAO code
2    WebService1.WeatherStnCode:= "PHNL" : Hawaii
3    WebService1.Invoke()
4
5    :Output the returned eather data
6    message attention  WebService1.WeatherStnCode
7    message attention  WebService1.LongName
8    message attention  WebService1.Longitude
9    message attention  WebService1.Latitude
10   message attention  WebService1.Temperature
11   message attention  WebService1.Skies
12   message attention  WebService1.RelHumidity
13   message attention  WebService1.Time
```

The added infrastructure to support the Web Service Wizard includes the following classes:

- **Unisys_WebAppSupportLib:** An external class configured with all the necessary entry points for the WebAppSupport Service.

- **Unisys_WebAppSupportHelper:** A class that simplifies Web Service implementations by consolidating the common processes for establishing and invoking a Web Service.

- **WebAppSupportLibraryBuffers:** Contains the specially configured buffers for invoking the WebAppSupport entry points.

- **Unisys_WebAppTypes:** All types associated with the WebAppSupport library. >>
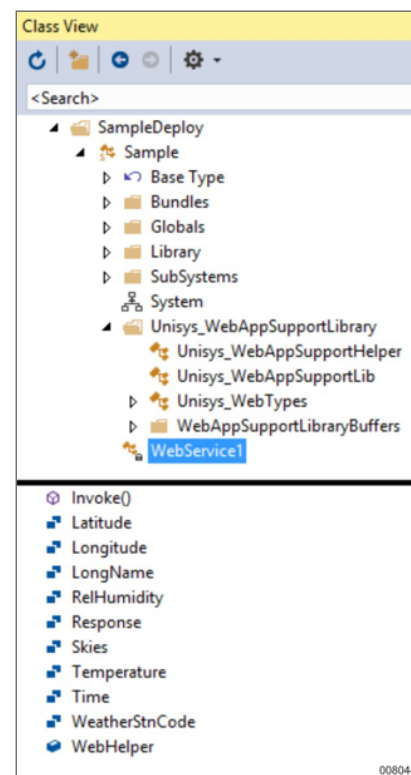
The above wizard not only provides ready-to-run Web Service access from your AB Suite Model, it also creates all of the necessary infrastructure to support Web Service calls using WebAppSupport.

Using the "WebService1" scenario captured in the image above, you now have a blueprint for calling other RestFul Web Services.

The Weather Service example above has a fairly simple interface (signature), so LDL+ coding for access to similar Web Services is a reasonably straightforward task.

For example, calling out to the Weather Service from an AB Suite report simply requires the following logic:

```
 1     :Invoke the Weather Service using a valid ICAO code
 2     WebService1.WeatherStnCode:= "PHNL" : Hawaii
 3     WebService1.Invoke()
 4
 5     :output the returned eather data
 6     message attention "Station Code = "      & WebService1.WeatherStnCode
 7     message attention "Long Name ="          & WebService1.LongName
 8     message attention "Longitude = "         & WebService1.Longitude
 9     message attention "Latitude = "          & WebService1.Latitude
10     message attention "Temparature = "       & WebService1.Temperature
```

However, when the interface has many arguments, it does make manually coding the associated AB Suite class attributes a bit more complex. Fortunately, a future AB Suite release will provide enhancements to support the discovery and importing of Web Service class definitions.

In the interim, you can use Business Integrator software as an alternate means of executing this part of the process. All you need to do is create a new Web Service Access Component. And you only need to enter enough information to allow the Web Service "class" to be created.

Here's how.

Creating a Web Service Access Component

Business Integrator provides a wizard you can use to create a Web Service component specification. >>
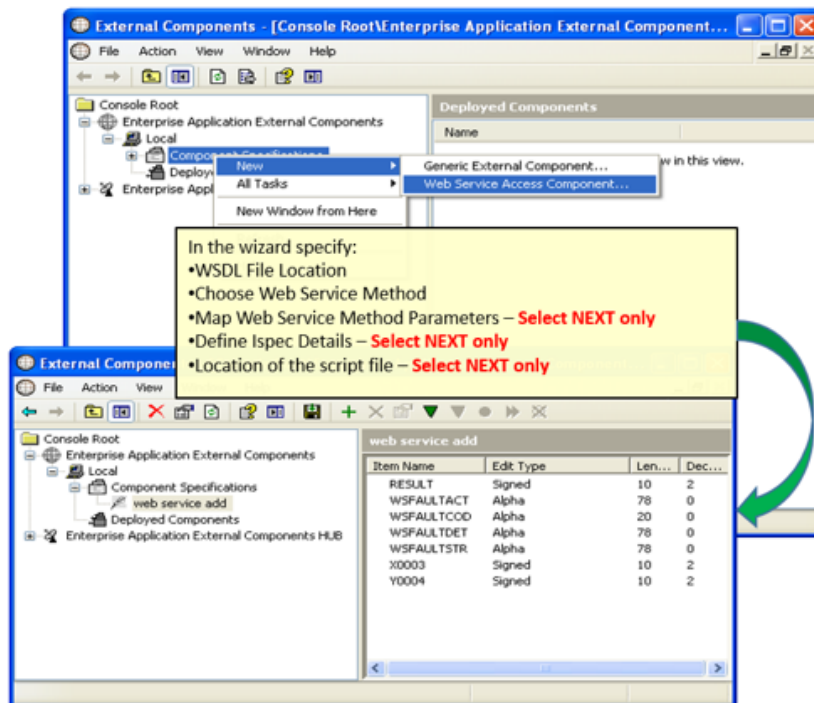
To create the specification, you need to:

1. Right-click the "Component Specifications" node.

2. Select New>Web Service Access Component. A wizard will be displayed.

3. Complete the necessary information and proceed through each page of the wizard.

4. Follow the above steps to create a .Model file, using Action>Extract. Next, in "Save as Type," select the .Model type for AB Suite.
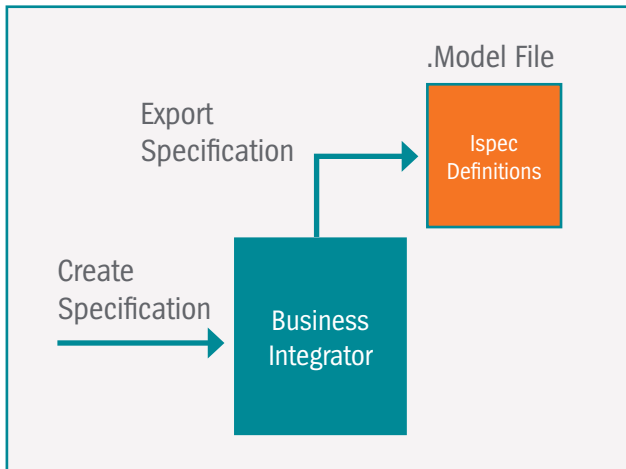
Once you complete these steps, move over to AB Suite, import the model, and give it the Segment name "NEW OWNER."
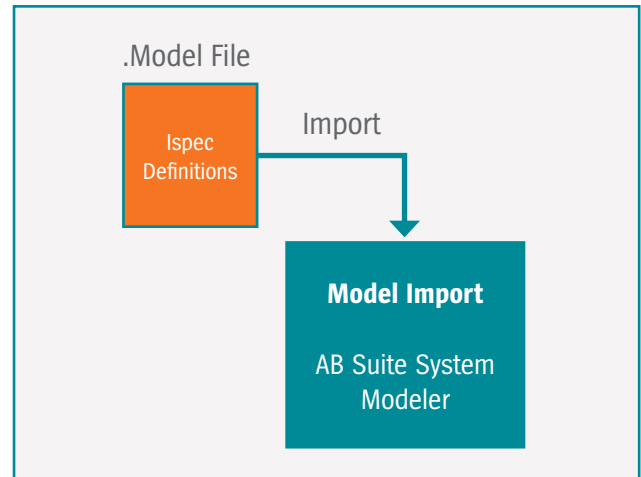
As shown above, this will result in an Ispec with the structure, layout, and format of the required Web Service. But since only the Ispec's class will be used for the Web Service call attribute definitions, the stereotype can be removed. This new "Web Service" class can now be modified to include an INVOKE method, which calls the various WebHelper methods – using the same approach and order as the "WebService1" scenario above – and returns the results of the call into its local attributes. Note that the "INVOKE method" name was chosen to maintain consistency with the wizard's provided structure.

Strictly speaking, WebAppSupport supports RESTful Web Services. However, this doesn't preclude you from using it to call SOAP Web Services. To do that, you'll just need to provide some additional LDL+ Logic for crafting the XML request as a SOAP request, and then parsing the XML SOAP response.

Please note that the XML Parser for ClearPath MCP must be installed and configured on the host MCP system to call Web Services. Please refer to the WebAppSupport documentation on the Unisys support site to learn more.

Microsoft Azure and VSTS: An Agile Business Suite How-to Preview

By Rob Henrichs, IT Specialist, Unisys

Note: *This article documents some of our observations from an early proof-of-concept project we conducted using an implementation model that is not currently supported/is not qualified for use with the current release of AB Suite. Although we are currently evaluating ways to support your use of AB Suite with these capabilities, Unisys makes no promises as to when or if it will release a final commercial version of the product supporting the implementation model discussed in this article. Unisys also makes no representations about using a cloud environment, and you and/or your organization alone must decide, whether using a cloud-based development environment is right for you, according to your requirements, policies, and the circumstances (including legal and regulatory rules) affecting your use.*

Working in the cloud has become commonplace these days. And since many Agile Business Suite clients have reached out asking how they could develop in the cloud, we decided to investigate the possibilities of using Microsoft Azure and Microsoft Visual Studio® Team Services (VSTS) with AB Suite Developer and the AB Suite Runtime for Microsoft Windows.

Azure is a cloud service for building, testing, and deploying applications through a network of Microsoft managed data centers. Since AB Suite applications are developed using Microsoft tooling and can be used in a Windows runtime environment, testing in the Azure cloud made perfect sense.

In fact, an Azure Windows VM is reachable via an RDP connection – just like any other VM – it's just located in the Azure cloud.

Building Your VM

Creating a VM in the Azure Dashboard is a simple, wizard-guided process. You start by defining the main properties and selecting the amount of processors, memory, and disk space it will have. Once you do, your VM will be automatically provisioned. >>

After creating the VM, the dashboard will give you options for starting, stopping, and connecting to it. Select the "connect" button, and you'll be given an RDP link you can use to connect to the VM exactly as you would with any other server using an RDP client.

Adding AB Suite to the VM

The easiest way to get the required AB Suite software onto the VM is to use the option in the Windows RDP client to connect local drives. Or, you could use an Azure storage account and upload the software that way. Once that's done, you can connect to the storage using a mapped disk.

When the necessary software is copied to the VM, you simply install AB Suite as you normally would.

Testing the VM

We tested this process by installing AB Suite 6.1 Developer and the Windows Runtime in an Azure VM. The Azure VM we created for the test with was configured with 2 vCPUs, 16 GB memory, and 130 GB disk space.
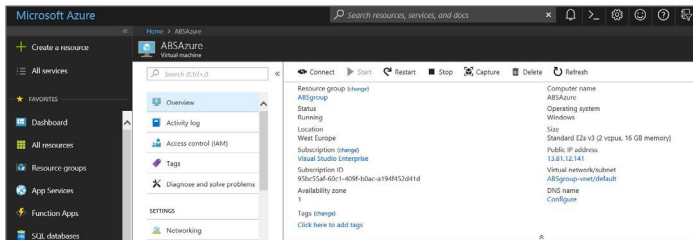
We did a local installation, with all of the necessary Microsoft SQL Server®, Visual Studio, and AB Suite software on the same server. The installation process was very simple and straightforward.

One of the primary reasons for testing AB Suite Developer with Azure was to see how it would work with the cloud-based Microsoft Team Foundation Server (TFS) service available in VSTS.

To test this, we used a sample Model we had already loaded into VSTS – see the following image – and then connected to VSTS from the Azure VM. This allowed us to automatically build a local Model database from the local VSTS workspace, and then use it to test debugging on a local SQL Server database and check in changes to VSTS.
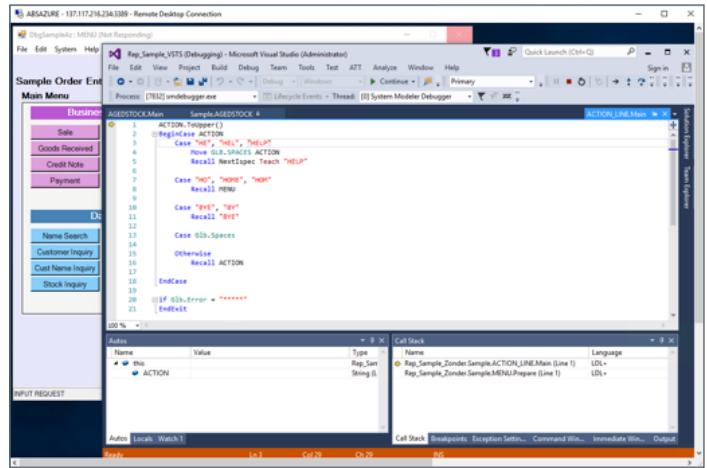
Using VSTS in this way allows for multiple Azure VMs to be built for and shared by your developers, much in the same way you would with local machines and TFS.

In addition to using Debugger, we also investigated deploying a local AB Suite on Windows runtime environment to the same Azure VM as the AB Suite Developer software. This behaved as expected, as the process is essentially no different than building to any local machine. To test access, we connected to the deployed system via a local WinForm client. >>
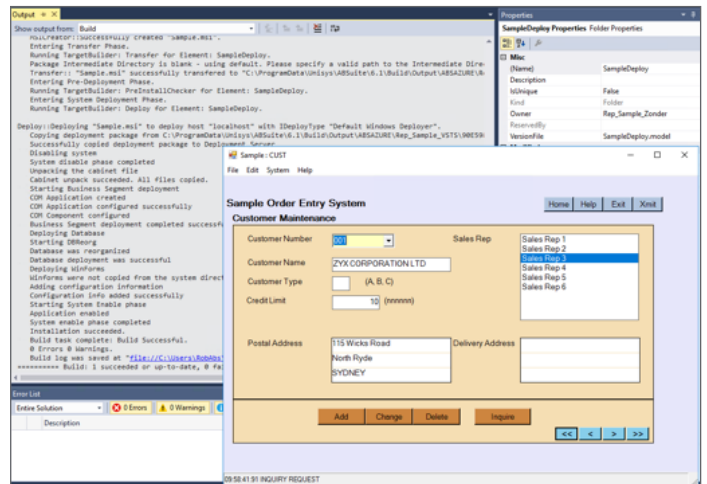
Getting Started

Creating and setting up an Azure VM for use with AB Suite Developer is straightforward and very similar to setting up any standalone laptop, PC, or VM environment. The only real challenge is getting the necessary files, like the installation media, on the VM. Once you complete this step, using VSTS makes spinning up development environments a simple step you can execute without investing in local hardware.

The Azure VM behaves as any other VM server, with no special requirements for AB Suite. When using an Azure VM as a standalone development and test environment, it's essential to use some form of version control, and VSTS is by far the easiest. It's TFS in the cloud, after all.

To get started in VSTS, you'll first need a Microsoft account. If you don't have one already, you can create a VSTS account here.

After you create an account and log in for the first time, you'll see a web interface that's very similar to the one used in TFS.
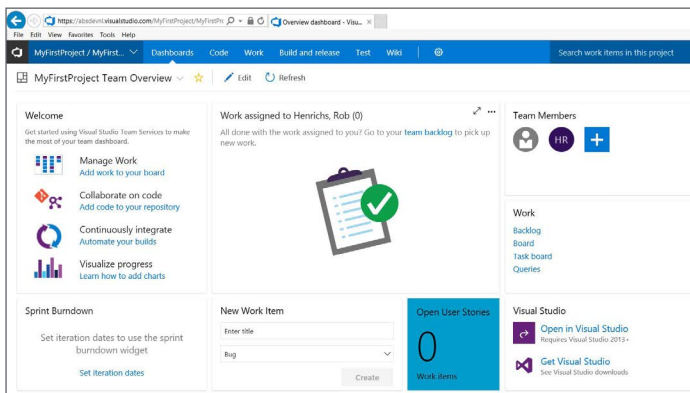

To connect in Visual Studio, go to the "Manage Connections" option in the Team Explorer and enter the VSTS created URL. Or, if you're on the same server, use the link from the web interface.
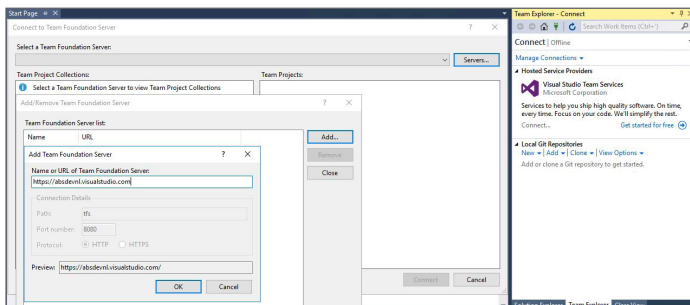
After that, open the solution in TFS and AB Suite will request a name for the model database name to be used, just like when connecting to TFS.
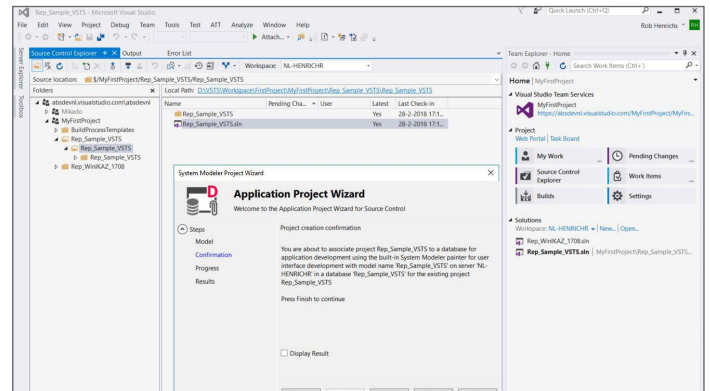
Finish this step, and you're ready to start using VSTS source control, including the Checkout and Checkin functions, just as you would in TFS.
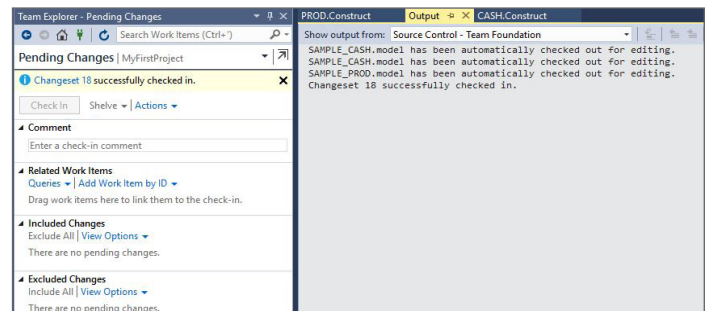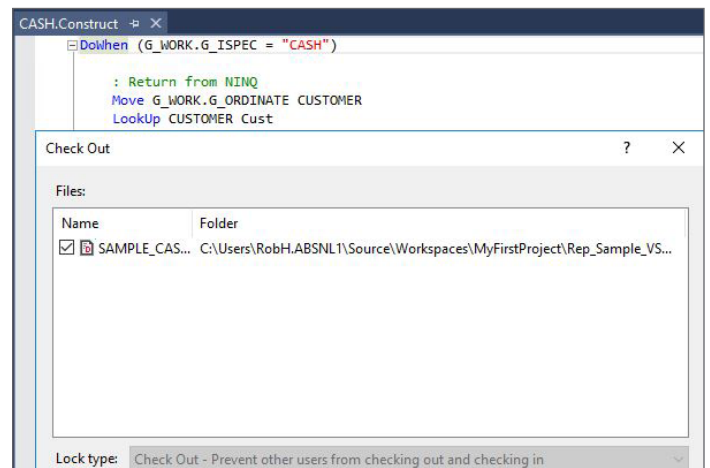
If you'd like to explore the powerful combination that is AB Suite and VSTS in your own development environment, or if you'd like to learn more about the capabilities described in this article, please email us at ABSuite@unisys.com. We'd be happy to help!

Sharpening Your Skills with AB Suite Education Services

Whether you're a former EAE user that's new to the Agile Business Suite environment, or a long-time AB Suite expert, one thing is certain: it's never too early – or too late – to sharpen your skills.

With this in mind, Unisys provides a number of training courses as part of our AB Suite Education Services offering. Though we've created a robust, standardized curriculum, you're free – and encouraged – to adapt this set of course material to your organization's specific needs and skill mix.

Please note that because the AB Suite development environment operates as a plug-in to the Microsoft Visual Studio IDE, some experience with Visual Studio is highly recommended. There are numerous online resources to help build your team's Visual Studio knowledge, as well as training from a number of sources.

When it comes to learning the basics of AB Suite development, the following courses represent the core AB Suite training curriculum.

Agile Business Suite Developer for EAE Users (CEL8022)

This course is for technical staff who will be developing applications within AB Suite. While we developed course material specifically for clients moving from EAE to AB Suite, the title "for EAE users" should not discourage those without EAE experience from attending.

We've increasingly been training users who have minimal familiarity with model-driven development right alongside those with considerable EAE experience in this course. In fact, a mixed class often makes for a more productive, powerful training experience. This is a perfect opportunity to infuse additional application development skills into your team, while increasing synergy across your development staff.

This class is typically four days in duration.

Maintaining and Enhancing AB Suite Applications (CEL8031)

This course is intended for technical staff who already have some experience using AB Suite in their own real-world development processes.

This instructor-led course provides additional guidance and best-practice recommendations around creating models in System Modeler and generating AB Suite applications. You'll spend time learning the various concepts and tasks involved in application development using AB Suite and its associated environments and modelling constructs, as well.

This course also focuses heavily on student-instructor interaction, specifically around students' experiences with AB Suite to date. We also conduct mentoring sessions with participants, so they can clarify any points of confusion or concern and gain more information on various topics.

This class typically runs between eight and ten days, depending on the experience level of the attendees and any specific needs you might wish to address.

Using Team Foundation Server (TFS) Source Control with AB Suite (CEL8034)

This instructor-led course targets technical staff involved in development and release management, who already have some AB Suite experience. While the course primarily focuses on using TFS for source control, we'll also explore leveraging TFS for build and test activities. >>

In addition, we'll offer instruction around installing, configuring, and implementing TFS in the AB Suite development environment through a combination of classroom instruction, exercises, and pre-recorded demonstrations.

This class is typically three-and-a-half days in duration.

These core courses are just a small sampling of the rich AB Suite training content we offer. Please contact your Unisys representative or email us at ABSuite@unisys.com to explore how you can tailor an AB Suite training plan to your organization's needs.

New additions to our libraries of How To documents, white papers, and other useful information include:

- **How To:** How to Upgrade Models Under Source Control from AB Suite 4.0/5.0 to AB Suite 6.1 (**NEW**)
- **How To:** Configure Firewalls for Networked AB Suite Windows Runtime (**Updated**)
- **Software Qualification and Support Matrix:** AB Suite 5.0 (**Updated**)
- **Software Qualification and Support Matrix:** AB Suite 6.1 (**Updated**)

To view these and other resources, simply go to public.support.unisys.com and choose "Documentation" in the "Public Information" box located on the left-hand side of the screen. No special login is needed.

In addition, there are several pieces of thought leadership available on the AB Suite homepage:

- AB Suite in the Application Lifecycle
- Agile Development with Agile Business Suite
- Unisys Agile Business Suite: Capitalize on Change, Don't React to It

We also encourage you to view the list of available AB Suite training courses. A blend of instructor-led and computer-based trainings, these great educational resources include graphics, interactivities, simulations, and demonstrations with voice-over narration.

To stay up to date on the latest happenings in the ClearPath Forward world, please subscribe to the ClearPath Forward Connection newsletter – and give the August 2018 issue a read.

And to learn about everything the ClearPath Forward Services portfolio has to offer, please visit our web site and check out our brochure.

If you're looking to explore AB Suite on an evaluation basis, please download AB Suite Express today! This free download includes the full AB Suite package – AB Suite Developer and AB Suite for Windows Runtime – as well as a "getting started" course designed to help you begin exploring everything AB Suite has to offer in no time.

UNISYS | Securing Your Tomorrow™