Issue 20 | May 2010

UNISYS
imagine it. done.

Developing Agility

A Newsletter for Unisys EAE and Agile Business Suite Customers

Contents

Not a subscriber of Developing Agility?
Don't miss the next issue – sign up in the eCommunity

Q&A with Europaeiske

EUROPÆISKE
REJSEFORSIKRING A/S

We recently spoke to Flemming Voigt, IT Manager at Europaeiske, about the company's move to Agile Business Suite. Located in Denmark, Europaeiske is a specialized insurance company that provides coverage to Danish travelers. Read on to learn about the IT environment at Europaeiske, and to get Mr. Voigt's thoughts on the main benefits he hopes to realize from Agile Business Suite.

Developing Agility (DA): Please tell us about the IT environment at Europaeiske.
Flemming Voigt (FV): We have one EAE (Enterprise Application Environment) application, which supports our core travel insurance business, and is divided into separate logical systems. One manages policy administration, including issuing new policies, extending policies, and terminating policies. We also offer an insurance booking engine as a web service that has been integrated into several software suppliers' solutions, including the travel business in Denmark and Sweden. The application also manages commission handling and calculation for the travel agencies selling our products to end users.

We also use the application for calculating and posting insurance tax, generating payments to claimants, and delivering consolidated and cleaned transaction data to our data warehouse, which runs on a Microsoft® Windows® server.

The EAE application consists of 410 Ispecs and around 600 Reports.

At the time we migrated, we were running the application on a ClearPath Libra Model 300 Server with 30 MIPS. In January 2010, we installed a ClearPath Libra Model 450 Server with 40 MIPS, which is the smallest version you can get. This was not because the Libra Model 300 was too small, rather we wanted to have a disaster recovery machine that could run the full production load, so the Libra Model 300 was installed in a remote backup center where it acts as cold standby for the current production machine.

We have two EAE/Agile Business Suite developers and five .NET developers who do all the graphical interface development, including web front ends. Internally, we have around 100 users accessing the core application, but since we are selling insurance over the Internet – where all the business logic is developed in LDL and all data is stored in our DMSII databases – we actually have 1,000-plus users connected to the system. Over the last few months, we have handled more than 200,000 transactions each day.

DA: What business benefits do you expect Agile Business Suite to help you realize?
FV: In terms of business benefits, we expect to continue to be able to leverage our investment in our EAE applications and use them in our IT environment into the future. In the meantime, we have also been using LINC/EAE/Agile Business Suite to reuse our logic – first with terminal interfaces, then with graphical clients, then as web pages, and finally as web services.

DA: What about the technical benefits?
FV: The expected technical benefits are of a more strategic nature. We want to be able to attract new talent by going to an environment that is well known to .NET developers.

In addition, we expect some of the object-oriented (OO) features that are made available in Agile Business Suite, specifically methods, to make it easier for us to maintain our applications in the future and build new functionality using OO techniques. >>

DA: What features of Agile Business Suite influenced your decision to make the move?
FV: The primary influencer was that Agile Business Suite is integrated with the Microsoft Visual Studio® platform. We were also attracted to the version control tool, new OO programming features, and the safe-passage migration.

DA: You went live on Agile Business Suite 1.2 in November 2009. What have been your experiences using Agile Business Suite in a production environment?
FV: We have not seen any difference in the way the MCP runtime is handled, so from an operator point of view there have been no changes. We have actually not observed any changes in runtime performance, but the response time on online transactions is a little better, though the difference is quite small.

DA: What level of support did Unisys provide before, during, and after you went into production?
FV: During the migration project we had support form Unisys Netherlands, which was backed up by the support team in ACUS to handle any issues we discovered during the migration and verification process.

DA: What do your developers think about Agile Business Suite Developer?
FV: There is no doubt that working in Visual Studio is quite different than working in EAE, but after a couple of weeks, the developers got used to Agile Business Suite. And, the experience they gained during the workshop using some of the new features in Agile Business Suite has helped them become more familiar with the new environment.

DA: Did you modernize the graphical front end of your EAE application as part of this project?
FV: Since we have been a heavy user of Component Enabler and have built many front ends ourselves, the interfaces did not change during migration. This meant that our end users actually had no knowledge of the migration, since everything worked as it had before we moved to Agile Business Suite.

The actual migration of the runtime environment involved just doing a full build of the model, which meant that we had only 10 to 15 minutes of downtime while database reorganization removed some internal structures that are not used by the Agile Business Suite runtime environment.

DA: Tell us what your developers have to say about the new Agile Business Suite toolset.
FV: We did a three-day basics course, just to get everyone used to working in Visual Studio, and then we took the Agile Business Suite CBT.

Developing report frames really took some time to understand, as did working with the Painter in Visual Studio, and we have actually found that things were better in these areas in EAE. It is very difficult to get an overview of a frame when a lot of attributes are included.

Writing logic was the first thing we got used to, and at the moment we are close to the same level of efficiency we experienced when we moved from EAE.

From an EAE developer's viewpoint the real benefits should come in the form of easier knowledge sharing with our .NET developers.

DA: You had a detailed project plan that included one week of onsite support and training. How did you manage to keep it so compact?
FV: Actually, we started the project in June 2007 with a one-week workshop that introduced us to Agile Business Suite, but due to some problems generating our EAE model at that time, the project was closed down. >>

When we had the workshop in beginning of September 2009, with the same Unisys instructor as in 2007, we experienced a déjà-vu moment because he knew our environment so well, which made it a very effective week that helped us prepare for all the activities that needed to be done during the real migration.

DA: Several consultancy activities were done off-site – how do you evaluate this type of arrangement? Do you know how much time the migration cost you and how many days Unisys was involved in the project?
FV: Since we knew the Unisys people from the workshop, it was a really effective way to do these steps at the time we were ready, without having to invest time and money in travel.

In addition, the remote support tool, which allows us to follow as a Unisys instructor works on our desktop, helped us feel very confident throughout the process.

We contracted Unisys for around 200 hours of consultancy, but Unisys also invested a number hours in learning the best ways to integrate Microsoft Visual SourceSafe® (VSS) with Agile Business Suite. In addition, we invested around 450 hours of developer resources to manage the migration and testing of the migrated system.

DA: What are your plans for new development with Agile Business Suite?
FV: There are no plans to build new systems using Agile Business Suite as of today, but with a system of the size we are supporting, we are adding new functionality to the existing system all the time. Our business has been used to getting new functions implemented in short timeframes. Actually, we are only planning projects one quarter ahead, so we are able to adapt to new business opportunities as they emerge.

Many thanks to Flemming Voigt for describing his organization's experience with Agile Business Suite.

Unisys Services Add Value

As Flemming Voigt mentions in the Q&A, Unisys services personnel worked with Europaeiske at key points in the project. Developing Agility checked in with two of the consultants to get their perspective on what made the project a success.

"*In critical projects, such as Europaeiske's move to Agile Business Suite, good planning is essential, and special consideration must be given to timing and the impact on the customer's live environment. And, the plans must cover activities performed by Unisys and by the customer. In regards to this project's success, excellent communication and agreement at all levels concerning what was to be done, when and by whom, turned out to be important factors.*"

– Floor Goedemondt, Project Manager, TCIS, Unisys Netherlands

"*At the Unisys Test and Performance Center, we imported Europaeiske's source and tested various version control and release management scenarios. Afterwards, I spent a week onsite where I conducted training in the morning and spent the rest of the day helping with the migration. The week went very smoothly and we made good progress on training and migration. Not long after, Europaeiske performed the final migration of their system themselves and it was completed without any problems worth mentioning. So in my opinion, this was a very successful project!*"

– Ching Hong Lam, Consultant, TCIS, Unisys Netherlands

UFSS: An EAE Application Modernization Case Study

Unisys Financial Services System (UFSS) provides a fully functional, low-risk, and secure end-to-end banking solution for the servicing of mortgages and savings. Used by a number of financial institutions in the United Kingdom, UFSS is well regarded for its ability to maximize productivity, reduce risk, and contain costs while delivering a consistent and fast customer experience to address the highly complex processing requirements of this industry.

UFSS is built using EAE and, like most EAE applications, it has been refined and extended over the years. Today, UFSS represents the intellectual capital of more than 20 years of domain experience and is proven to be a flexible, compliant solution that provides a reliable, efficient, and cost-effective platform for originating and servicing mortgages and savings.

In recent years, the Unisys UFSS development team has updated the solution to provide access to transactions via the Internet. And more recently, it has modernized the application to allow UFSS to consume a Web Service and expose UFSS functions as services. This case study reviews the business drivers for the changes and recent steps the UFSS team has taken to further modernize this business-critical application.

Customer Requirements

The current version of UFSS (affectionately known as UFSS Classic) is a screen-based solution comprising more than 2,000 individual Ispecs. Experienced end users have no problem navigating through the application using the five-character Ispec identifiers. But for new users, the learning curve can be significant – and there's little guidance as to how a user should string together the right set of transactions in the proper sequence to accomplish a given task.

Through conversations with its customer community, the UFSS team identified the need to help banks and building societies:

- Enable business processes that lead the end user through the right steps in the right order to complete a business transaction according to current bank policies
- Bring new users up to speed quickly and cost-effectively – whether they are novice users or experienced personnel who are new to the UFSS application
- Ensure that customer interactions are performed efficiently and consistently, leading to compliance with the UK's Financial Services Authority mandate to "treat customers fairly"
- Reduce costs throughout their operations

Given the significant value in the UFSS solution – from a financial investment and business operations perspective – it was important to find a way to evolve the existing application to address these requirements. >>

Solution: A Process-driven User Interface

The UFSS development team had already completed significant modernization work on the application, including building a framework that allows the solution to participate in a service-oriented architecture (SOA). Attendees of the EAE Track at the 2009 UNITE Technology Conference may have seen Tony Bradford's presentation about this work (visit the eCommunity to review Tony's slides). As a part of this project, the team created a Host Integration Gateway, or HIG, that allows EAE applications to invoke Web Services and outside services to submit a request for data from an EAE Ispec. As a part of this project, the team developed the concept of a service Ispec – one that is stateless and does not rely on GLB.Work to store session data.

To address customer requirements for improved ease of use and better enforcement of standard processes, the UFSS team realized the answer lay in providing a new layer of process automation that leverages the existing solution. Using UFSS application logic and a new Business Service Layer, the team created the concept of Process-driven User Interfaces (UIs) that consolidate and simplify multiple transactions into a single or very few screens and do so in a prescribed, yet data-dependent, order. This approach enables the presentation of a business process flow that guides a user through a complete financial transaction and also ensures the process is followed correctly and consistently. The Business Service Layer can also communicate with other applications behind the scenes – providing access to services and data, which is incorporated into this same UI.

Key technical attributes of the Process-driven UI approach include the following:

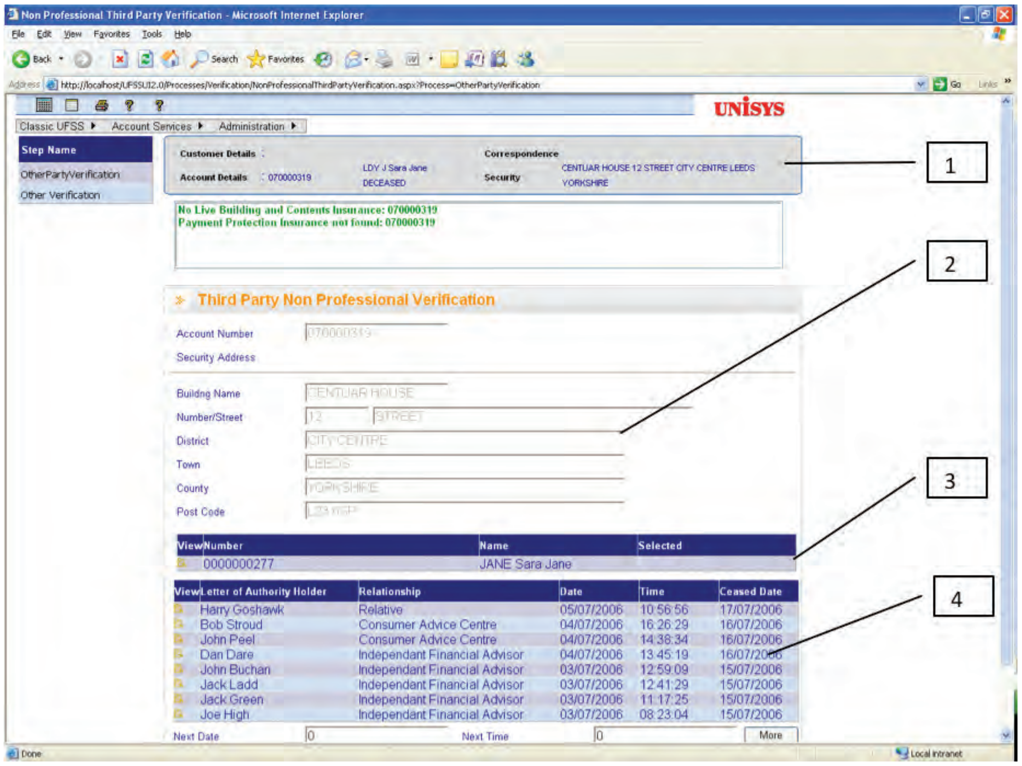
- Process-based screens are painted manually using ASP .NET technology.
- AJAX controls are invoked as needed to speed up the display of data as the user navigates across a screen – further improving the overall user experience.
- In the background, asynchronous calls are made from the UI Layer to UFSS services through a Transport Layer that interfaces to Component Enabler. Results from these service calls are used to populate data on the screen as the user steps through a process (i.e., from one field to the next).
- Different fields are displayed on a screen depending on the data entered in previous fields.
- Because the Process-driven UI uses sitemaps, customers are able to implement .NET role-based security to further control end-user access to certain functionality. >>

How the Process-driven UI Works

As is often the case, looking at a few samples helps to illustrate exactly how a technical concept, such as the Process-driven UI, works.

Sample 1: Retrieving Data

This screen gathers the required data by calling various UFSS services, which are performed asynchronously and automatically paint the screen.


Non Professional Third Party Verification - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/UFSSUI2.0/Processes/Verification/NonProfessionalThirdPartyVerification.aspx?Process=OtherPartyVerification
UNISYS
Classic UFSS Account Services Administration
More
Done
Local intranet
1
2
3
4

Step Name
OtherPartyVerification
Other Verification

Customer Details	:		Correspondence	
Account Details	: 070000319	LDY J Sara Jane DECEASED	Security	CENTUAR HOUSE 12 STREET CITY CENTRE LEEDS YORKSHIRE

No Live Building and Contents Insurance: 070000319
Payment Protection Insurance not found: 070000319

Third Party Non Professional Verification

Account Number	070000319
Security Address	
Buildng Name	CENTUAR HOUSE
Number/Street	12 STREET
District	CITY CENTRE
Town	LEEDS
County	YORKSHIRE
Post Code	L23 9SP

View	Number	Name	Selected
	0000000277	JANE Sara Jane	

View	Letter of Authority Holder	Relationship	Date	Time	Ceased Date
	Harry Goshawk	Relative	05/07/2006	10:56:56	17/07/2006
	Bob Stroud	Consumer Advice Centre	04/07/2006	16:26:29	16/07/2006
	John Peel	Consumer Advice Centre	04/07/2006	14:38:34	16/07/2006
	Dan Dare	Independant Financial Advisor	04/07/2006	13:45:19	16/07/2006
	John Buchan	Independant Financial Advisor	03/07/2006	12:59:09	15/07/2006
	Jack Ladd	Independant Financial Advisor	03/07/2006	12:41:29	15/07/2006
	Jack Green	Independant Financial Advisor	03/07/2006	11:17:25	15/07/2006
	Joe High	Independant Financial Advisor	03/07/2006	08:23:04	15/07/2006

Next Date 0 Next Time 0

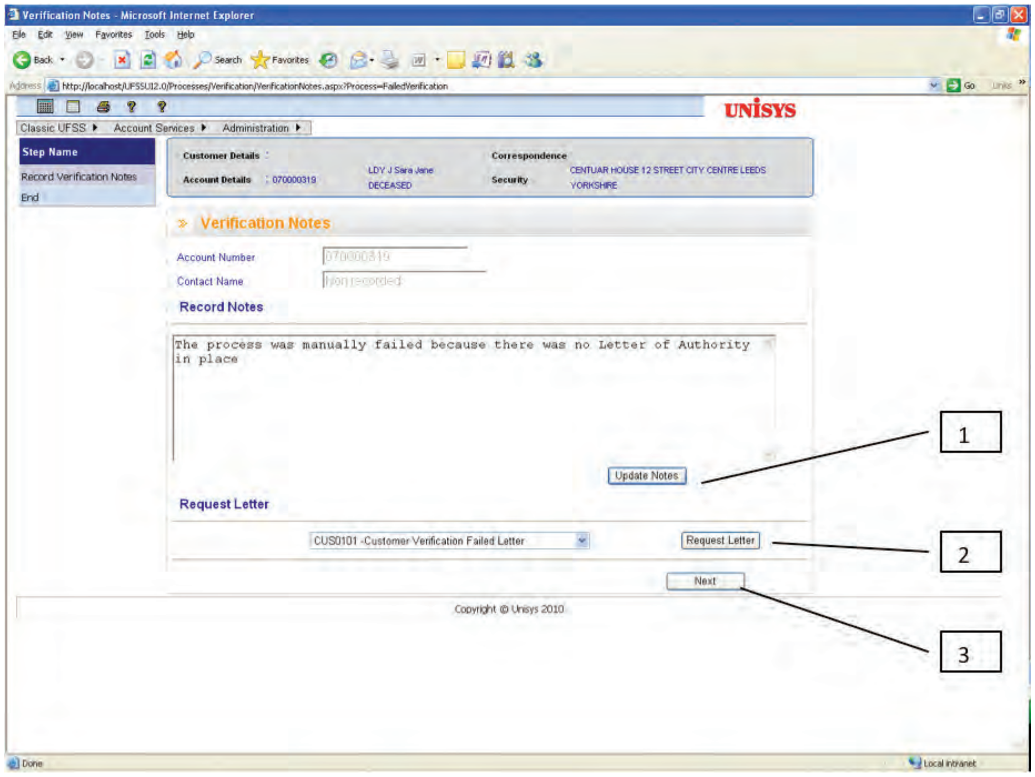
1. The context data is provided from the GetAccount service (SRV61).
2. The Security Address is provided from the GetAccountDetails service (SRV35) via a .NET Web user control that is placed on this page – and is reused on other pages, as well.
3. The Account Holders are obtained from the GetAccountHolders service (SRV20), which is another reusable .NET Web user control.
4. The Letter of Authority list is obtained from the GetLetterOfAuthorityHistory service (SRV09). The image button on the left-hand side allows a row to be selected, which, in turn, invokes another service to obtain the Letter of Authority details.

As illustrated in Sample 1, UFSS presents table data as a drop down list. The information is delivered from the Transport to the UI layer in such a way that it can be bound to the data grid with no C# code. The number of rows is configurable and controlled by the "numbers of rows" attribute on the grid, which is passed into the service Ispec as a parameter.

A separate render method is used to automatically update the screen data items from the Ispec model data. This is accomplished by ensuring that the names of the screen controls match the Ispec, which also reduces coding efforts. A similar method is used for combo boxes, check boxes, and radio buttons. >>

Sample 2: Sending Data

In Sample 2, data is sent to UFSS depending on the button a user clicks.


Verification Notes - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/UFSSU12.0/Processes/Verification/VerificationNotes.aspx?Process=FailedVerification
Go Links
UNISYS
Classic UFSS
Account Services
Administration
Step Name
Record Verification Notes
End
Update Notes
Request Letter
Next
Copyright © Unisys 2010
1
2
3
Done
Local intranet

Customer Details	:		Correspondence	
Account Details	: 070000319	LDY J Sara Jane DECEASED	Security	CENTUAR HOUSE 12 STREET CITY CENTRE LEEDS YORKSHIRE

Verification Notes

Account Number: 070000319

Contact Name: Non recorded

Record Notes

The process was manually failed because there was no Letter of Authority in place

Request Letter

CUS0101 -Customer Verification Failed Letter

1. The Update Notes button invokes the SRV11 service (another reusable .NET Web user control) to update the account scratch pad. Once pressed, the control is disabled to prevent multiple updates.
2. Request Letter calls the SRV06 service to request a letter to be sent to the customer. As with Update Notes, this is a .NET Web user control and once the button is pressed, the control is disabled to prevent multiple requests.
3. The Next button requests the next step in the process. In this example, it completes the process and returns the user to the starting position.

While existing UFSS logic is leveraged for the Process-driven UI, the team had to develop some new service Ispecs, which are stateless and act as black-box services to be called. Once UFSS customers migrate to Agile Business Suite, the UFSS developers will be able to take advantage of its ability for an Ispec to invoke another Ispec – thus avoiding the need to have two versions of a transaction: one for the Classic UI and another for the Process-driven UI. >>

Results: Proven Business Logic Enables New Processes

The UFSS team recently demonstrated the results of this new application modernization project at its User Forum, and customers were genuinely excited by the prospect of what the Process-driven UI will deliver. They immediately saw the operational efficiency the approach provides by eliminating the need to have experienced users involved in user acceptance testing, as well as the usability benefits and reduced training costs for novice users. What's more, the UI is of great interest to UFSS prospects, who see the value of an intuitive user interface that is based on the actual flow of the business process and are impressed with the modern look and feel of the application.

The following processes have been automated and are scheduled to be deployed in a live customer environment shortly:

- Account Search
- Customer Search
- Account Summary information
  - Account Holders
  - Account Warnings
  - Transaction Summary
- Customer Verification
- Switching service to allow Process and Classic to integrate seamlessly

More processes will be automated as time, budget, and customer requirements dictate. General release of the UFSS Process-driven UI will happen in 3Q2010.

As for the UFSS development team, the project has brought its EAE and .NET developers closer together by virtue of their joint effort to present a better UI while using the proven logic in the EAE application.

If you have any questions about the information presented in this case study, please contact Tony Bradford, Solution Architect, Unisys TCIS CET UKMEA: tony.bradford@gb.unisys.com.

UNISYS
imagine it. done.

Important Changes Coming to Client Tools

By Diane McGonigle, Agile Business Suite Program Manager, Unisys TCIS

Historically, Unisys has provided a separate set of Client Tools for both EAE and Agile Business Suite. Each set includes a number of client generators that enable developers to create a wide variety of end-user interfaces, as well as the runtime infrastructure needed to deploy them. While there are a few differences between the content and capabilities of EAE Client Tools and Agile Business Suite Client Tools, there are also many similarities. In addition, multiple versions of the same generators exist within each set of Client Tools, either EAE or Agile Business Suite.

While there are benefits to providing many different options, it can be confusing for even the most experienced EAE and Agile Business Suite developers. Sometimes, there really can be too much of a good thing. Over the coming months, Unisys will be implementing some changes that make our Client Tools easier for developers to use and provide the same functionality for EAE and Agile Business Suite applications. These changes will also enable us to bring new technologies faster to you in the future.

By the end of 2010, the following plans will roll out:

- Evolution to a single, common set of Client Tools for EAE and Agile Business Suite. This change means that either of the Agile Business Suite release 1.2 or 2.0 Client Tools can be used with EAE applications. Either configuration works today – and has for a number of years. More significantly, future Client Tools enhancements will be incorporated into Agile Business Suite releases only and will work with EAE applications.
- Focus on the C# versions of the client generators and a phasing out of support for the Java-based versions. An EAE Interim Correction (IC) release in June, 2010, will introduce required changes to Enterprise Application Developer that enable it to call any of the C#-based generators.
- Discontinuance of support for generators that rely upon unsupported technology (e.g., the Microsoft Visual Basic® 6 and SOAP Toolkit Web Services Generators). Replacement capabilities are available today in Agile Business Suite Client Tools for both of these generators.
- Discontinuance of support for ASP Generator. More modern technology is available with the ASP .NET WebForms Generator AND it includes support for AJAX and mobile devices.

Common Client Tools (fully effective 31 December 2010)

- ASP .NET WebForms Generator
- Visual Basic .NET Generator
- Presentation Client
- ASP .NET Web Services Generator

After finishing the work, the EAE Client Customization Kit, which is a separately licensed product, will contain the final versions of the ASP, Visual Basic 6, and SOAP Toolkit Web Services Generators for any developers that wish to continue to use these tools without support.

The two runtime interfaces in Component Enabler – Java and .NET – will continue to be supported. If you have a custom generator that uses the Java runtime interface, you will not need to change how you deploy the generated clients. We are discontinuing support of the Microsoft JVM (Microsoft ended support a number of years ago). We will continue to support the Sun Java Runtime Environment (JRE). >>

We will also continue to support the interfaces in Enterprise Application Developer and AB Suite Developer that allow you to use Java-based client generators, as well as the newer C#-based generators. So, if you have a custom client generator that is written in Java, you will be able to continue using it with EAE or Agile Business Suite.

For More Information

You probably have some questions. Many more details about these changes can be found in our “Client Tools Changes for Enterprise Application Environment and Agile Business Suite” white paper. Developers that are using any of the EAE or Agile Business Suite Client Tools should download the document from the Unisys Support web site or the eCommunity.

In addition, this issue’s Engineering Corner article examines various scenarios to help you evaluate how these changes will impact your environment. In many cases, there will be minimal or no impact.

If, after reviewing these two sources, you still have questions about how these changes will impact your organization, send an email to ABSuite@Unisys.com.

Engineering Corner: Client Tools Next Steps

By John Papachristos, Client Tools Technical Lead, GTC Australia, Unisys TCIS and
Nagendra Purushotham, Team Lead, GTCI, Unisys India

With the upcoming changes in Client Tools, affected EAE and Agile Business Suite customers must take action to ensure their client interfaces continue to function. This article outlines various scenarios and offers guidance on what steps to take to bring your EAE and Agile Business Suite applications up to par.

Scenario 1: Using the ASP Generator

If you are currently using the ASP Generator and you have not customized your generated clients in any way, your move to ASP.NET Web Forms Generator is straightforward. Simply make a configuration change to the Component Enabler Bundle property to specify the ASP.NET Web Forms Generator, regenerate your form definitions[1], and deploy the ASP.NET Web Forms-based clients to your IIS Web Server. Since these forms were designed using the EAE or Agile Business Suite Painter, to your end users they will look and feel exactly the same as before.

Now, if you use the ASP Generator and you have **customized** the generated client in any way, you have some work to do. That's because you'll have to recreate these modifications in the new ASP.NET Web Forms environment. Naturally, this will require familiarity with the replacement technology and programming language, i.e. ASP.NET and C#. Once those changes are made, the same steps noted above should also be followed to invoke the ASP.NET Web Forms Generator and deploy the new clients.

By current standards, the ASP Generator is ancient technology, and the ASP.NET Web Forms Generator offers a nice set of **additional capabilities** that EAE and Agile Business Suite developers will want to take advantage of, including:

- AJAX support
- Mobile device support
- Scripts for easier deployment
- Dynamic attributes
- Automatic combo box support, which ASP does not generate by default
- Customization options using the Component Enabler WebFormRenderer
- Automated Test Tool support (with Agile Business Suite 2.0 only)
- Panels and improved button group capabilities (these features are available with Agile Business Suite only because they require capabilities supported by the screen painter in Agile Business Suite Developer)

Scenario 2: Using Visual Basic 6.0

If you are currently using the Visual Basic 6[2] Generator, your logical migration path is to Visual Basic .NET. If no customization of the Visual Basic 6 forms has been made, the switch to Visual Basic .NET is simple. Update the Component Enabler Bundle property to specify the Visual Basic .NET Generator, regenerate your form definitions[1], and deploy the Visual Basic .NET clients to your end users' workstations. There will be no change to the look and feel for your end users.

Customers who plan to switch to Visual Basic .NET must install the Microsoft .NET Framework on their end users' workstations – Visual Basic .NET clients require this software. Customers who are running Visual Studio 2005 or later will be able to take advantage of the ClickOnce publishing option to automate installation of all the necessary client software components on end-user machines in their network.

If you have **customized** the generated Visual Basic 6 client, you will have to do some work to apply these modifications in the new environment, which requires skills with Visual Basic .NET and the .NET Framework. Follow the steps noted above to complete your transition to Visual Basic .NET. >>

1 Note that if you've made changes beyond the configuration change to the bundle property, you may have to regenerate your EAE or Agile Business Suite application. Otherwise, just regenerate the bundle.
2 Microsoft Visual Basic 6 is not supported by Microsoft and hasn't been for more than two years.

The Visual Basic .NET Generator offers **additional features** that EAE and Agile Business Suite developers can use, including:

- Scripts for easier deployment
- Dynamic attributes
- Customization options using the Component Enabler WinFormRenderer
- Automated Test Tool support (with Agile Business Suite 2.0 only)
- Panels and improved button group capabilities (these features are available with Agile Business Suite only because they require capabilities supported by the screen painter in Agile Business Suite Developer)

Scenario 3: Using the SOAP Toolkit Web Services Generator

If you are using the SOAP Toolkit Web Services Generator, you will move to the ASP. NET Web Services Generator. As described in Scenario 1 above, the change is simply a matter of a configuration change to the Component Enabler Bundle to specify the new generator, regeneration of your Web Services using the ASP.NET Web Services Generator, and deployment to the IIS Web Server.

Because we are talking about creating Web Services from EAE or Agile Business Suite applications, you have to consider the potential impact of this change on the consumer of your service. For example, ASP.NET Web Services employ newer structures, referred to as complex types, which were not frequently created by the SOAP Toolkit Generator. This means that the application that invokes or consumes the Web Service may need to be modified to allow for retrieving and setting properties in complex types. These considerations will be further documented in a How-To that will be available when the Client Tools changes are released.

EAE and Agile Business Suite developers will appreciate the improved set-up and deployment options that come with the ASP.NET Web Services Generator, as well as the potential to better secure Web Services using Microsoft's WS-Security infrastructure. And, the package will continue to include the Discovery tool, which aids the testing process for Web Services.

If you are using Business Integrator to access Web Services based on the SOAP Toolkit, which is subsequently replaced with ASP .NET Web Services, you will need to update your component specifications to reference the new ASP .NET Web Services. In most cases, you will also need to modify the component specification script logic to handle the complex types defined in the ASP .NET Web Services.

Scenario 4: Using the Java Version of a Generator, Moving to the C# Version

As noted in another article in this issue of Developing *Agility*, the Java versions of the Client Tools generators will eventually be phased out. Therefore, you should plan to move to the C#-based version at some point in the coming months. Agile Business Suite users can make the change immediately. EAE customers will be able to make the change with the implementation of IC 3270, which introduces support for the C#-based generators and will be available in June 2010.

This switch to a C#-based generator is a simple change in the configuration properties in the Component Enabler Bundle folder. Instead of specifying the Java class name for the Java-based generator, enter the required C#-based generator assembly name, e.g. replace com.unisys.jellygen.GenerateFormASPdotNET with GenerateFormASPdotNET.dll.

Note: In Agile Business Suite System Modeler, a drop-down list of the standard C#-based generators will become available to make this change easier.

A benefit of moving to C#-based generators from Java-based ones is that your standard development environment will no longer require installation of Java (JRE and SDK), unless you wish to generate a Java-based client or have a custom Java-based generator. >>

Scenario 5: Using a Java-based Custom Generator

If you use a custom Java-based generator, you can continue to use it as we are not eliminating the Component Enabler generate environment for Java or the Component Enabler Java runtime interface. It should be noted that we do not provide support for custom generators, unless there is evidence that a particular problem lies with our interface or client environment.

Scenario 6: Using Custom Clients NOT Created Via a Client Tool Generator

Some customers have written custom clients for their EAE applications – without using Client Tools. These clients will continue to function without change because the upcoming Client Tool changes do not impact the Component Enabler Runtime interface.

Stay Tuned for More Information

We are preparing a series of How-To documents that provide more details regarding the steps involved in migrating to the newer client generation technologies. In addition, the Component Enabler User Guide explains how to use all the generators, including ASP.NET Web Forms and Visual Basic .NET Generators.

If you have any questions about the information in this article or how these changes impact your organization, send an email to ABSuite@Unisys.com.

Farewell to Microsoft JVM

Microsoft stopped supporting its JVM product several years ago and this environment will no longer work with Client Tools by the end of 2010. Customers using the Microsoft JVM should make immediate plans to use the alternative interfaces that we support.

- If you are currently using the Microsoft JVM to generate Component Enabler client applications, you should switch to the Sun JRE now. You should also plan to move to the C#-based generators by the end of 2010.
- If you are using the Microsoft JVM at Runtime, you should plan to use the Sun JRE or the .NET Runtime interfaces, depending on the technology used by your client application.

Calendar

What	Where	When
UNITE Annual Technology Conference	Baltimore Marriott Waterfront Hotel, Baltimore, MD	May 23-26, 2010