What are some good techniques to convert an Ms Access application to a .Net Application?

Asked 16 years, 3 months ago Modified 5 years, 10 months ago
Viewed 10k times

7

We have a 12-year-old Ms Access app that we use for our core inventory warehousing and invoicing system. It IS already running on an SQL Server backend, but all the "logic", forms and reports are in Access. After experiencing the massive amounts of maintenance sludge it took to turn inventory transactions from non-temporal to temporal, I realized that I need to someday convert this thing into code so I can better manage the logic in a much more maintainable and testable environment.

What are some techniques that would allow me to convert it into a .Net application in a manageable and efficient manner?

One idea was to convert the queries to stored procedures, then convert the app into an Adp project.But I am still clueless as to how to handle the forms and reports.

Also, I am the only developer for my company, if that matters.

ms-access legacy

Share

Improve this question

Follow

asked Sep 19, 2008 at 20:07

Gilligan

1,515 ● 2 ● 15 ● 16

5 Answers

Sorted by: Highest score (default)

6

Short answer: the migration doesn't seem like something easily automated.

My guess is that your best bet is to rewrite (and install) the system one piece at a time, even if (perhaps) it forces your users to run the old and new versions side-by-side for a while to use different bits of functionality. You can minimize that hassle by careful consideration of which features to migrate and in which order.

For example, you might have one user whose job role requires him or her to use just one screen all day. If you migrate that screen first with accompanying functionality, that user can be on the new system immediately and leave the old one behind, reducing your maintenance load.

So those are just some ideas based on not too much information. I hope this helps anyway.

Share Improve this answer

Follow

answered Sep 19, 2008 at 20:22

catfood

4,331 ● 5 ● 32 ● 57

5

As you already have asp.net with some business logic you could open this up to access as a web service (asmx files). Google for the Microsoft Office Web Services Toolkit for your version of access (xp/2003 etc.) and this will write vba proxy classes for you to call the web service. You can bind web service data to the forms through code (vba to read and write to controls) or create local temp tables with data from the web service and use regular access binding.

Depending on what you are most comfortable with (code/tsql) you can put logic in stored procedures or in a business logic layer or hybrid (both). I find it easier to test code than stored procedures and like not being bound to sql server for business logic i.e. if you want to change the database or want to develop/test components offline without a database. New .net features such as LINQ have pretty good performance so you don't have to rely on stored procedures for database activities.

Keep the access front end user interface until you have refactored all your business logic/data access to web services. You can then create an asp.net app that consumes the web services or a winform app if you want.

(Stay clear of wpf, as a ui, for the time being as it is a steep learning curve and doesn't yet have a datagrid that can compare to the access datasheet view.)

Reports

The access reports can can be upsized to sql server reporting services (vba in reports doesn't upsize and it is better to write some tsql in stored procedures). If you don't have the full sql server product you can still use the reportviewer control to write you reports (see http://www.gotreportviewer.com/) in asp.net (or winform with the standard version or up of Visual Studio) binding to ado.net datasets.

Other options: You can write .net dlls and use com interop. This approach allows you to start writing functionality gradually. Don't use .net ui e.g. a winform as it won't play nicely with access ui. You could write business logic or data access logic and then call these classes from vba. You can then move this code to asp.net or web services if required.

Things to rule out:

I don't like the approach of writing a new app with side by side versions. As a single developer you have enough to worry about. You will probably end up adding features in both versions and debugging two versions rather than one.

The vb6 forms interop does not work for access.

ADP as stated is pretty dead. (I never liked them as I often use local tables to optimize performance and they can only be called through code and not linked)

You may be able to convert your vba modules and class modules to vb.net using The Visual Basic Upgrade Wizard (in visual studio) but it doesn't upsize everything (e.g. dao/ado code to ado.net code) and doesn't create code that is optimized for .net and may not be easy to write unit tests on depending on the design of the vba code. I recommend rewriting the code (try Test Driven Development if you are serious about testing to see if you like it).

Share Improve this answer

Follow

answered Oct 16, 2008 at 12:26

Simmo

224 ● 2 ● 4

YEs, I use Test-Driven Development exclusively (slowly bringing our spaghetti website under control). I am glad to hear about the Report conversion and I like the com interop idea to at least get the complex bit under control. – Gilligan Oct 16, 2008 at 12:42

2

I would consider looking at the Interop Forms Toolkit. As I understand it, this tool makes it quite easy to use .NET forms from within VB6, so perhaps it can also be used from within Microsoft Access? If so, it may help you migrate the application to .NET in an incremental fashion. Doing a quick search, I was unable to find any guides on

using it with Microsoft Access, so I apologise if this turns out to be a blind alley.

Share Improve this answer

Follow

answered Sep 19, 2008 at 20:45

Paul Batum

8,405 ● 5 ● 42 ● 47

1

Converting to an adp will not be a good solution in the long term - this technology is abandoned by Microsoft.

If you want to switch to .net (why? do you have a reason to favour .net?) I suggest you start some reading, try to create some simple apps and then start the task of converting this database to an application.

But...

I think you and the company need to think about the risks involved in this project. What will happen if you get sick, just in the week that management needs some reports that don't already exist? I would suggest that you seek a small local software development company, they will be glad to help you. Maybe you can arrange that you continue to be the 'lead developer' and only use them for back-up.

Share Improve this answer

Follow

answered Sep 20, 2008 at 13:27

Birger

4,353 ● 23 ● 35

Interesting idea. I would choose .Net because I am already maintaining our website, which is in Asp.Net. So I would in essence integrate the business logic from the inventory system into our current infrastructure. SOme of the business logic is already duplicated in both places. – Gilligan Sep 20, 2008 at 18:29

0

I have a similar problem, and addressed it by creating a versioned deployment system in the Access front end (grab and extract a CAB file), figuring out the required AppDomain manipulation to be able to load the correct CLR version into the Access process, load a .config file, and post data **both ways**.

It uses standard C DLL calls, so no COM registration required, but also sadly no Unicode support.

Send Command - Generic, I could have done everything with this. Open Form - Intended to be a "drop in" replacement for DoCmd.OpenForm Open Report - Intended to be a "drop in" replacement for DoCmd.OpenReport

So make a new report, or migrate an existing one to SSRS, make the format standard, and then change DoCmd.OpenReport to netDoCmd.OpenReport in Access.

Follow a naming convention to know where the reports are to load from, and a standard method for pulling the data needed for the report.

Now I am migrating one form or report as capacity permits, or when a change is requested to it.

Because who can stop feature development for a year to do it all in one hit?

MDI doesn't work properly though. I think there's some work I still need to do around SetParent and UPDATE_UISTYLE

All of this makes the UI in process and in window with Access. I build everything in DLLs, and the final step will be to create an EXE that loads the "first form" and use that to replace the Access front end.

Share Improve this answer

Follow

answered Feb 15, 2019 at 9:13

Roger Willcocks
1,659 ● 13 ● 28