

Should I bother learning Windows Workflow?

Asked 15 years, 8 months ago Modified 10 years, 7 months ago

Viewed 1k times



4

Okay, so I've learned WPF and WCF and that was great. I didn't bother with CardSpace because it's probably not gonna take off.



Should i bother with Workflow,..is it worth it? What's the big deal?



.net

workflow

Share

Improve this question

Follow

asked Mar 31, 2009 at 16:09



[EightyOne Unite](#)

11.8k ● 14 ● 83 ● 107

8 Answers

Sorted by:

Highest score (default)



7

Ever seen [Leon Bambric's post on Workflow software?](#)

Share Improve this answer

Follow

answered Mar 31, 2009 at 19:16



[George Mauer](#)

122k ● 139 ● 395 ● 626



That's a funny post. However, in my organization, just having the workflow designer was helpful because it allowed non-programmers to easily disable certain activities without being confronted by code. So the code is still written by me, but if necessary, others can make very small modifications easily without accidentally breaking something else. – [Dave](#) Jan 12, 2010 at 18:28



6

It is worth learning or at least understanding WF. WF offers a different paradigm for building apps, a paradigm which *may* become mainstream in the future.



Enterprisey? Yes, maybe today. But longer term WF or at least the paradigm it promotes, may become ubiquitous and universal. Using Workflows to describe programs may today be in the stage where OO and C++ was, 20 years ago.



Learning a new thing broadens your thinking. For example, even if you use C# or Java in your day-to-day job, knowing Lisp helps broaden your thinking about how to solve problems. IMO. WF is the same.

A second thought is to think like an investor. If you want to raise the value of YOU as a programmer, whether to learn WF would depend on the market or job you are in. WF skills in your space or in your company may or may not be worth your extra effort.

**3**

Windows Workflow is useful for several different kinds of application. If you're not writing one of them, then there's no need to bother learning it.

1. Human workflows - obvious
2. It can be useful for complex UI navigation
3. It can be very useful as a way of processing several operations in parallel. It's a very easy way to do asynchronous processing.
4. Long-running operations consisting of multiple steps
5. Rules-based processing
6. Any situation where you need to permit your customers to customize behavior. You can ship custom activities that pertain to your application, and customers can then combine and parameterize them.

But if you don't need any of this, why bother? Just read some of the whitepapers, or look at some of the videos and see if any of it looks interesting.

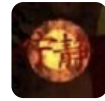
One interesting thing that's not obvious: both the workflow designer and the rules designer can be hosted in arbitrary Windows Forms applications. This allows them to be

reused in your own application, if, for instance, you need a way to customize conditional processing.

Share Improve this answer

answered Mar 31, 2009 at 16:34

Follow



John Saunders

162k ● 26 ● 249 ● 402



3



I think the general consensus on the current Windows Workflow is that the product is not quite ready for primetime yet. If you want to get anything more than trivial accomplished with the current framework, you will likely have to write a few custom tasks to fill in functionality gaps. I've also had issues with the designer flaking out when the complexity of the workflow starts piling up.

From what I've been hearing/reading about Windows Workflow, the next version (v4.0) is a complete rewrite that fixes a lot of the issues in the current version. They claim backward compatibility with v3/3.5 workflows, but you probably will have to know the framework pretty well to know what changed and how to fix it if it doesn't work right away.

Unless you have a project that absolutely demands this functionality today, you're probably better off waiting until the new bits ship.

Share Improve this answer

answered Mar 31, 2009 at 17:12

Follow



Scroll Lock

186 ● 5



1

Like Paul said, it depends on the environment you work in. Personally, since I never know where life is going to take me (corporate development, startup, etc.) I try to learn everything I can when I have the chance.



My advice would be, if you have the time, the intuition, and the development environment...learn it. You never know when it might come in handy at some point down the road.

Share Improve this answer

answered Mar 31, 2009 at 17:07

Follow



Justin Niessner

245k ● 40 ● 414 ● 544



0

Thanks for all the advice guys. Cheeso get's the answer since he got the most up votes but all opinions are appreciated. I think I'll take Scroll Lock's advice and wait until v4.0 until I take a serious look.



Share Improve this answer

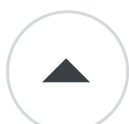
Follow

answered Apr 1, 2009 at 8:21



EightyOne Unite

11.8k ● 14 ● 83 ● 107



I originally really liked the concept of workflows, but in the end I think I'll just use something like stateless or roll my

0

own statemachine, and use a database for persistence.



My biggest issue with it right now is that there still seems to be a huge memory leak in the workflow runtime. Makes me kind of nervous.



Share Improve this answer

answered Jan 12, 2010 at 18:30

Follow



Dave

15k ● 13 ● 94 ● 146



-1

It depends on the type of software you build. WPF is neat. WCF has its uses. Windows Workflow is pretty "enterprisey", though. It requires more setup and can get complex.



Unless you build enterprise software that would benefit from a definable workflow, I wouldn't bother much with it.



Share Improve this answer

answered Mar 31, 2009 at 16:21

Follow



Paul Lefebvre

6,376 ● 4 ● 30 ● 36

"requires more setup"? Were you thinking of BizTalk? They're separate things, and unrelated. – [John Saunders](#) Mar 31, 2009 at 16:29
