



WHITE PAPER

## Microsoft Windows Workflow Foundation Comparison with Captaris® Workflow

By David Hanowski, vice president technology, Captaris, Inc.  
July 25, 2006

## TABLE OF CONTENTS

|   |          |
|---|----------|
| <b>SOFTWARE AND PLATFORM EVOLUTION.....</b>       | <b>3</b> |
| <b>WHAT IS WINWF?.....</b>                        | <b>3</b> |
| <b>WHAT IS CAPTARIS WORKFLOW?.....</b>            | <b>5</b> |
| <b>COMPARING WINWF AND CAPTARIS WORKFLOW.....</b> | <b>6</b> |
| <b>WINWF AND OFFICE 2007.....</b>                 | <b>7</b> |
| <b>CAPTARIS TECHNOLOGY STRATEGY .....</b>         | <b>8</b> |
| <b>SUMMARY.....</b>                               | <b>9</b> |

The purpose of this white paper is to provide a technical comparison of functionality between Microsoft's Windows Workflow Foundation (WinWF) and Captaris Workflow. There has been a lot of misunderstanding of what WinWF and Captaris Workflow currently provide "out-of-the-box." The intent of this document is to clear up any confusion between the capabilities provided by the WinWF services, and the functionality delivered in the Captaris Workflow product.

## SOFTWARE AND PLATFORM EVOLUTION

Software is still a relatively young field, one that continues to evolve and change at a very rapid pace. Technology advances, development methodology improvements, and even the means by which software is packaged and sold changes constantly. For example, in just the last ten years we have seen technology drive the normative software development paradigm from large, mainframe, green screen applications to distributed, PC-based, object-oriented, client/server applications and finally to (or some would argue, back to) host-based, service-oriented, client-agnostic applications.

In terms of platform technology the two biggest drivers in the past five years have been Sun's Java technology (and its many variants) and Microsoft's .NET. With its 1.0 release, Microsoft's .NET platform enabled products written with Microsoft tools to take advantage of concepts made popular by the Internet; concepts such as XML and Web services. These technologies were not directly demanded by the market. Rather, these enabling technologies evolved in response to what the market wanted from its computer systems. Business and consumer users wanted software applications that could integrate with each other, that were accessible across firewalls (i.e. B2C, B2B, etc.) and applications that were scalable and low maintenance.

XML and Web services were not the first technologies aimed at making it easier to integrate applications together. Indeed, over the last 20 years, many attempts to create integration protocols or frameworks met with varying—although modest—levels of success. Several standards definition bodies have delineated and promoted integration protocols, among the more successful was Electronic Data Interchange (EDI). The biggest problem with these protocols, however, was that they required a lot of custom software to be developed for each application.

The functionality provided by .NET 1.0 (and then 1.1) was compelling enough that most Microsoft ISV's quickly adopted .NET as their development platform. It provided the basic services required to facilitate the cross-application integration that the ISVs' customers demanded. Microsoft .NET was attractive to ISVs for other reasons too, but being able to process XML and Web services were seen as key to the future survival of any product, and .NET made it much easier to implement these technologies.

With each subsequent release of the .NET framework, more and more services were made available to the developer, allowing the developer to focus more and more on what their application was supposed to do, and less on the underlying platform services that are often likened to "plumbing". This is Microsoft .NET: A rich set of software services which enable application developers to build richer applications by allowing them to focus on their customer requirements instead of base system requirements. Microsoft .NET is not an application "product" as it does not deliver any complete end-user functions. Its services are put to use in end-user products by application developers.

The most recent addition to the .NET framework is Microsoft's WinWF—a central addition to .NET 3.0. The next section will describe how these "out-of-the-box" services provide more for the developer but still fall short of being a workflow "product."

## WHAT IS WINWF?

As Microsoft continues to "raise the bar" in terms of what they refer to as platform services, a compelling addition is basic workflow services. Providing services-level facilities to define, manipulate and execute a workflow model enables developers to add the concept of *process*—actively managing the steps required to address a business function—to their applications.

WinWF is a foundation. A "foundation" in .NET is a set of services which can be extended by the Visual Studio developer. So WinWF is just like the services in the .NET framework that enable XML or Web services integration. This is an important point because out-of-the-box, these services do not do anything specific for the end-user. If an IT System Administrator were to download and install WinWF, the end-users would receive no new workflow functionality. A developer needs to write low level code to extend these services to the applications or solutions for the benefit of the end-users.

WinWF lacks the user-level objects and features required by applications. Why? Because these features are specific to the process-enabling of a given application. WinWF contains no higher level objects such as *tasks*, with specific events associated with them. It has no concept of a *task list*, a *user* or *task timelines*. It does not include a default user interface. The database component of WinWF only stores minimal information about the workflow runtime. To use WinWF in an application, a developer must create a user interface so the end-users can complete their *tasks*. In addition, the developer would have to extend the persistence layer to store other runtime specific data that is important to the workflow.

Information that WinWF stores to the database is very basic and meant for maintaining data persistence. Storing any other metrics about the processes will require developers in Visual Studio 2005 to write custom code for their applications.

Although there are probably several scenarios under which WinWF might be used by developers, we expect the following two to be the most common.

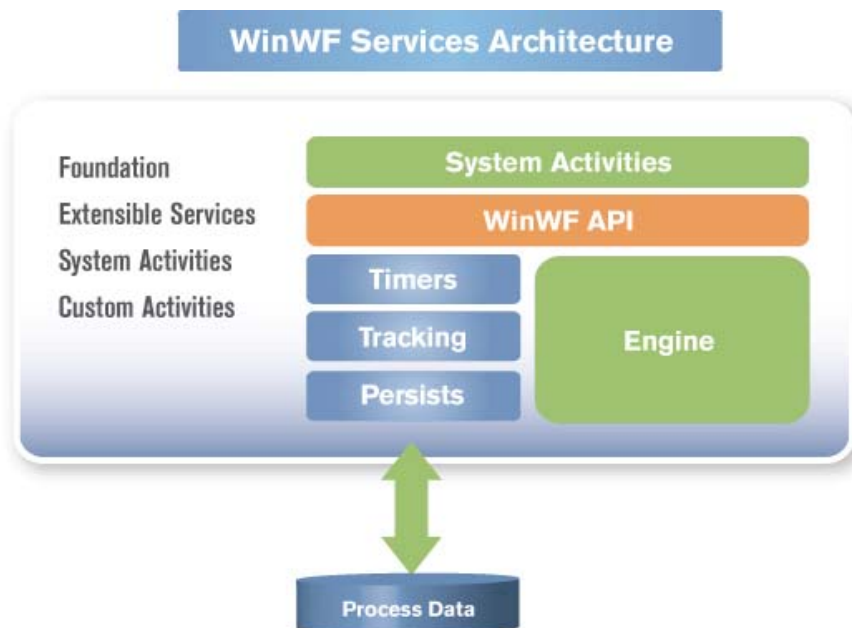
In the first scenario, the organization already has a partial solution for their business process automation problem, either the form of a custom-coded application or an ISV's vertical software product. In this scenario, the developer will use the WinWF services by building the *custom activities* required to integrate WinWF into their existing solution, perhaps in order to link together some existing data entry forms or to create some forms specific to their application. This could, for example, extend an existing ERP system to provide simple automation of purchase order approval. This would require some level of effort but it would be something a competent Visual Studio developer could accomplish.

In the second scenario, we expect some developers to choose to further extend WinWF by defining higher order objects such as *tasks* or *events*, and then decide to store the data associated with these higher order objects in a database. This effort would be significantly more difficult than the effort in the first scenario. Some developers may even choose to extend their development efforts further so that their customer could get analytical data about the efficiency of what has been automated and then provide the tools for the non-developer to build workflow maps and drag-and-drop specific pieces of functionality. This crosses the threshold into building their own workflow product, a project which undoubtedly would take several person-years. Defining the necessary underlying data schemas alone requires very specific expertise, and would take many person-months to complete.

Although, the singular term "developer" has been used, in reality both of these scenarios would likely require a team of developers.

Graphically, Microsoft WinWF can be represented by the diagram in Figure 1.

**Figure 1: Windows Workflow Foundation (WinWF) Services architecture**

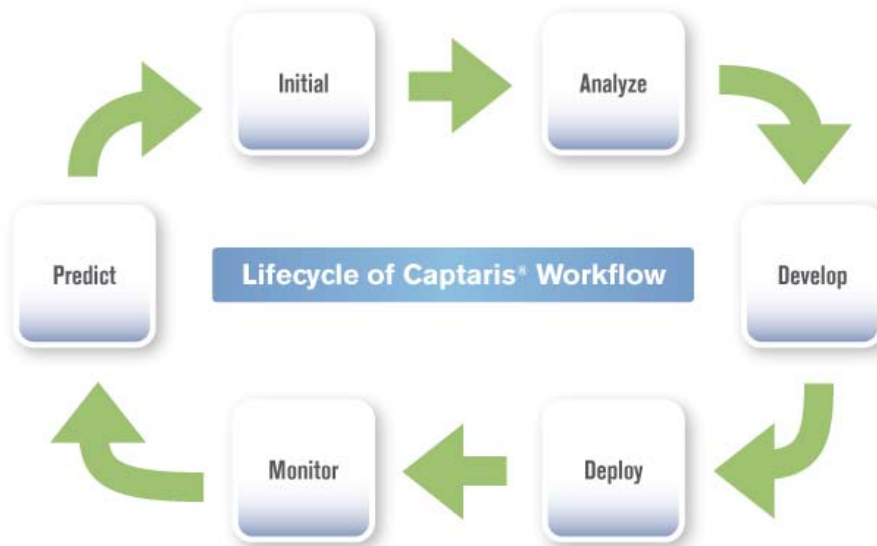


## WHAT IS CAPTARIS WORKFLOW?

Like Microsoft's WinWF, Captaris Workflow provides some core workflow services designed around a XML definition of a workflow model. However, the similarities end there. Core workflow services provide only the low-level foundation of a workflow "product." Captaris Workflow provides the higher-level objects, interfaces and tools so that end-users, business analysts, *and* application developers can rapidly create and deploy applications to manage complex business processes.

Captaris Workflow supports the users through the entire lifecycle of process-centric solutions. Code development is only one step—albeit an important step—in that lifecycle, as represented in Figure 2.

**Figure 2: The Lifecycle of a Workflow Solution**



Every lifecycle has phases and each phase has different stakeholders. Each phase is equally important for the successful deployment of workflow solutions. Captaris Workflow provides tools to assist the different stakeholders at each phase of the lifecycle.

- In the **Initial** phase, Captaris Workflow provides Return on Investment (ROI) and solution visualization tools to help the customer better understand their business problem and how our products can help them solve their business problem.
- As we move to **Analyze**, business analysis tools enable the modeling of high-level workflow structures such as tasks, routing layouts, business rules, default forms, document and image capture points and user responsibilities.
- In the next phase, we provide tools to **Develop** the solution. Captaris Workflow provides a rich workflow designer that is accessible to the Visual Studio developer. The developer uses this tool to write the underlying code required to complete the solution.
- After the solution is developed, the customer needs tools to **Deploy** the solution in a production environment. Typically, the solution is developed in the development environment then promoted to one or more test environments. Finally, the solution is promoted into production.
- After the solution has been deployed, additional tools (Business Activity Monitoring (BAM) and Knowledge Performance Indicators (KPI's) help business analysts **Monitor** the effectiveness of the automation of the business process.
- The last set of tools, help business analysts to **Predict** future events such as optimal resource management and solution bottlenecks. Metrics collected through business activity monitoring and other historical data are fed into simulation tools to assist with this predictive and modeling phase.

This is a simplified view of workflow solution development. In “real-world” projects the phases overlap, and there is often significant iteration between phases (i.e., there is often some “back and forth” between the **Analyze** and **Develop** phases). However, this should leave one with the understanding that core workflow services are not the “silver bullet” answer to building and deploying workflow solutions. Such services are critical, but only for one phase (**Develop**) of the lifecycle of a workflow solution. The more complex and industry-focused the business problem, the more time and effort is spent in the other phases of the solution lifecycle.

Perhaps the biggest opportunity for Captaris Workflow to improve is by “raising the bar” to the level where less and less technical expertise is required to **Develop** and **Deploy** complex workflow solutions.

In fact, it is the challenge of every product to simplify the functionality for the end-user it was originally intended to serve. ERP systems need to be simplified for the non-accountant and CRM products need to be simplified for the non-technical user. In a similar manner, we intend to evolve Captaris Workflow by providing simpler tools for the end-user or stakeholder responsible for executing each phase of the workflow solution lifecycle.

### COMPARING WINWF AND CAPTARIS WORKFLOW

Likely the best way to analyze the differences between Microsoft’s WinWF and Captaris Workflow is to examine the diagram in Figure 3.

Figure 3: Comparing Microsoft workflow offerings with Captaris Workflow



Figure 3 contrasts the components provided by Captaris Workflow (on the right hand side) with Microsoft offerings “out-of-the-box” with WinWF (the bottom left) and Microsoft’s own use of WinWF in SharePoint 2007 and Office 2007 (top left). The areas of functionality are segregated by Information Worker, Business Analyst, Developer and the core workflow engine or workflow kernel. Note, as previously mentioned, WinWF has a very basic database so the workflow can be persisted for a long period of time between activities. However it would have to be extended if you wanted to save custom runtime data.

Microsoft provides a very elegant integration with Visual Studio 2005 for the developer, as does Captaris Workflow. However, as tempting as it may be for the less experienced Visual Studio developer to jump in and start creating software to automate business processes, WinWF services are better suited to the experienced Visual Studio developer and still require significant effort to use effectively. The Captaris Workflow integration with Visual Studio can be an effective implementation for a wide array of developers from novice up through the highly experienced, and has proven itself in the field with this range of developers. In some cases, entire workflows can be created and deployed using Captaris Workflow without writing a single line of code, and leveraging the Captaris *Custom Action* framework.

Also, Microsoft provides very nice facilities inside of the Office 2007 product suite (including SharePoint 2007) that leverage the WinWF services. This will be discussed further in the next section.

However, the main software components that leverage the workflow engine and make Captaris Workflow a product simply do not exist with WinWF. Organizations that have true business process automation requirements—including the desire to measure and manage the effectiveness of the automated processes—would require a team of developers to build the infrastructure around the WinWF services. By way of analogy, this would be a similar effort to writing a custom CRM product on top of a database engine. Most organizations would choose not to write a custom CRM and we believe most organizations will choose not to write a custom workflow product.

## WINWF AND OFFICE 2007

Much of this document has discussed how, “out-of-the-box,” Microsoft’s WinWF is primarily meant for the Visual Studio developer and doesn’t really provide a lot of functionality for the end-user. Office 2007, on the other hand, has used the WinWF services to deliver some very nice “out-of-the-box” functionality.

Most of this functionality relates to the approval or rejection of a document, and more specifically, a Microsoft Office document (Word, Excel and PowerPoint). These are natural extensions of the collaboration facilities which are already offered by SharePoint.

SharePoint 2007 provides a number of workflow templates “out-of-the-box:”

- Document approval
- Collecting Feedback
- Collecting Signatures
- Disposition Approval
- Issue Tracking
- Consensus Approval

These templates can be extended via a wizard. Even so, they are meant for simpler workflow scenarios which involve Microsoft Office documents or InfoPath forms.

These standard and customized SharePoint workflows can be initiated from inside the Office 2007 authoring applications (Word, Excel, PowerPoint, etc.). The same type of review and approve scenario is the primary target for this workflow functionality. It remains to be seen whether or not the average user of Word wouldn’t simply just choose to email the document for feedback or approval rather than launch a workflow process.

## CAPTARIS TECHNOLOGY STRATEGY

The Captaris technology team has been engaged with the various Microsoft early adopter and technology awareness programs for WinWF, Office and SharePoint since mid-2004. This technology is by no means a surprise to us and our plan has been, and will continue to be, to embrace and extend any and all technologies which further enable us to reach the goals and to fulfill the expectations our customers have for our product. Being a “behind the scenes partner” to the development of this new generation of Microsoft workflow technologies has enabled us to influence some of the core WinWF services to our advantage. It has also given us deeper insight into new capabilities we can build in Captaris Workflow that will continue to provide value to our customers. One example of the latter is the Captaris Simulation, which we are now demonstrating with workflow models that have been created with WinWF services.

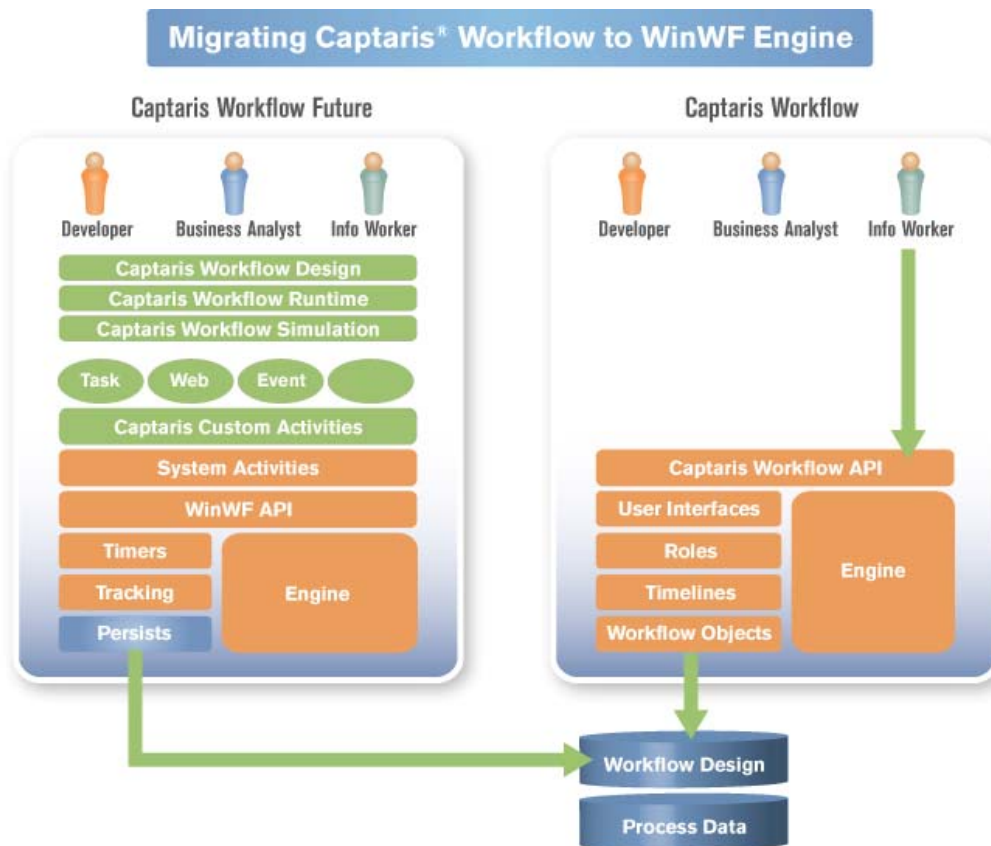
Our intent is this document will clarify that Microsoft’s WinWF does not by itself solve business problems. Even so, as Microsoft is using the WinWF services in Office 2007 there are many simple and generic workflow scenarios thereby addressed, more or less, “out-of-the-box.” Most of the workflow scenarios addressed in Office 2007, however, are *ad hoc* and isolated to small teams or workgroups.

So what is the opportunity for a product like Captaris Workflow? Does it make sense to go through an entire solution lifecycle as it has been outlined in this white paper for these types of workflow scenarios? For example, does it make sense to simulate, calculate ROI and analyze the effectiveness of simple document routing and approval? Probably not. However, we believe these simple and generic workflows addressed by Office 2007 will expose more people to the power of process enabling applications, and will encourage businesses to consider workflow enabling their higher value and mission critical applications specifically where integrations with Captaris RightFax® and Captaris Alchemy® are required. This is where Captaris Workflow shines.

Our intention is to fully exploit the services Microsoft is providing with WinWF. The Captaris product roadmap includes a release (code named *Hulk Hogan*, targeted for mid-2007) which will build on top of WinWF. This is mostly a technology decision as it simply doesn’t make sense to continue to write and maintain the low-level software services which someone else will do for you.

The best way to illustrate how we plan to utilize WinWF is to drill down to the workflow kernel level of the diagram used in the **Comparing WinWF and Captaris Workflow** section of this document.

Figure 4: Migrating Captaris Workflow to WinWF Engine



The first step in our plan is to develop the necessary software around the base WinWF interface. This “layer” of software, the Captaris *Custom Activities*, allows us to build the representations of the higher level objects in Captaris Workflow, the required building blocks of a rich workflow product. This will enable us to preserve much of the knowledge and intellectual property we have built over the years and to extend the functionality that Captaris Workflow provides to include a common name space for all Captaris products.

In addition, we will extend the persistence service which is responsible for storing workflow information to save all of the current data and business metrics which Captaris Workflow is currently tracking. This will allow us to preserve and even extend all of our efforts to date around capturing the metrics around business processes and the analytical reporting (BAM, Online Analytical Processing (OLAP) cube).

## SUMMARY

Microsoft’s WinWF has definitely introduced some confusion to the workflow technology landscape. In particular, the differences between what WinWF provides and what a product such as Captaris Workflow provides are being obscured. We hope that this document has clearly described the ways that the technologies differ, how their use models vary, and the ways that Captaris is and will use WinWF to the benefit of Captaris Workflow users.

WinWF raises the level of workflow services that are provided in the .NET platform. Whenever basic platform services are added, those products that leverage or overlap the functionality of those services need to raise their level of functionality. This is exactly what we intend to do with Captaris Workflow.

©2006 All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Captaris. The following are registered trademarks and trademarks of Captaris: Captaris, Alchemy, RightFax, Captaris Document Management, Captaris Interchange and Captaris Workflow. All other brand names and trademarks are the property of their respective owners. MC0010372