



WinFX



Windows Workflow Foundation

Parte 2/2

Paolo Pialorsi
paolo@devleap.it

DevLeap

<http://www.devleap.com/>

<http://blogs.devleap.com/paolo>

The DevLeap logo, featuring the word "devleap" in a bold, sans-serif font. The "v" is stylized with a blue checkmark and a small figure jumping over it. The background of the logo is white with a blue border.

Brevissima presentazione

- Sapete cosa facciamo
 - Consulenze, Conferenze, Corsi
 - Sviluppo ad hoc
 - Libri, articoli, multimedia
- Competenze
 - .NET Framework 2.0
 - SQL 2005 sia OLTP che BI
 - Mobile con VS 2005 e .NET CF 2.0
 - Software Architectures
 - .NET 3.0 (WinFx)
 - SharePoint



Agenda

- Workflow hosting
 - WF <-> Host communication
- Runtime Services
- Advanced Activities
 - Custom Activities
 - Conditioned Activity Groups
- Rules & Policies
- Designer Hosting
- Office 2007
- Q&A

Workflow Hosting

- Un workflow deve essere ospitato da un processo
 - Windows Application
 - ASP.NET Application
 - SharePoint
 - Office
 - Servizi Windows
- L'host si appoggia a WorkflowRuntime
- Il workflow “parla” con l'host tramite eventi e interfacce

Chiamare il Workflow dall'host

- Definire un'interfaccia
- Decorarla con ExternalDataExchangeAttribute
- Definire un evento
 - EventArgs: derivare da ExternalDataEventArgs
 - Rendere l'EventArgs serializzabile [Serializable]
 - Anche tutte le classi/entità utilizzate (ovviamente ...)
- Inserire nel workflow una HandleExternalEventActivity
 - Associare InterfaceType all'interfaccia
 - Definire EventName
- Implementare l'interfaccia (e l'evento) nell'host

Chiamare l'host dal Workflow

- Definire un'interfaccia
- Decorarla con ExternalDataExchangeAttribute
- Definire un metodo
 - Rendere serializzabili - [Serializable] – eventuali argomenti
 - Anche tutte le classi/entità utilizzate (ovviamente ...)
- Inserire nel workflow una CallExternalEventActivity
 - Associare InterfaceType all'interfaccia
 - Definire MethodName
- Implementare l'interfaccia (e il metodo) nell'host
 - Ricordarsi che la chiamata dal Workflow NON è thread-safe

Considerazioni

- Perché questo sistema di scambio/comunicazione?
 - Per rendere il Workflow indipendente dall'host
 - Potremmo implementare l'interfaccia in Windows Form e ASP.NET
 - L'interfaccia potrebbe essere un contratto WCF...
- Perché [Serializable]?
 - Perché in caso di persistenza dello stato gli oggetti devono essere serializzabili
- Perché Thread-safety?
 - Il thread di WF non è quello della form Windows Form
 - Windows Form richiede che interventi sull'interfaccia avvengano dal thread principale della UI
 - Quindi Invoke, se InvokeRequired

Workflow Runtime Services

- Il WorkflowRuntime può usare dei servizi
 - Classi derivate da: WorkflowRuntimeService
- Alcuni servizi nativi sono:
 - WorkflowLoaderService
 - WorkflowPersistenceService
 - SqlWorkflowPersistenceService
 - WorkflowSchedulerService
 - DefaultWorkflowSchedulerService
 - ManualWorkflowSchedulerService
 - TrackingService
 - SqlTrackingService
 - ExternalDataExchangeService
 - WorkflowCommitWorkBatchService

Persistenza

- I Workflow non “girano” 24x7
 - Possono durare molto tempo
 - Possono essere sospesi dal flusso (sleep o wait external event)
 - Possono girare su più server in NLB
- Durante le attese non possono restare in memoria
- Serve un sistema per salvarne lo stato e ricaricarlo all'occorrenza
- Si parla di “Persistenza dello stato”
- Aiuta ad ottenere elevata scalabilità

Persistenza dello stato in SQL Server

- Esiste un servizio nativo di WF:
 - SqlWorkflowPersistenceService
- Configurare il DB in SQL Server:
 - C:\WINDOWS\WinFX\v3.0\Windows Workflow Foundation\SQL\EN\
 - CREATE DATABASE SqlPersistenceService
 - SqlPersistenceService_Schema.sql
 - SqlPersistenceService_Logic.sql
 - Requisiti:
 - MSDE/SQL2000
 - SQL 2005 (Express)

Struttura dati

- Programmability
 - Stored Procedures
 - System Stored Procedures
 - dbo.DeleteCompletedScope
 - dbo.InsertCompletedScope
 - dbo.InsertInstanceState
 - dbo.RetrieveAllInstanceDescriptions
 - dbo.RetrieveCompletedScope
 - dbo.RetrieveExpiredTimerIds
 - dbo.RetrieveInstanceState
 - dbo.RetrieveNonblockingInstanceStateIds
 - dbo.UnlockInstanceState

Column Name	Data Type	Allow Nulls
uidInstanceID	uniqueidentifier	<input type="checkbox"/>
state	image	<input checked="" type="checkbox"/>
status	int	<input checked="" type="checkbox"/>
unlocked	int	<input checked="" type="checkbox"/>
blocked	int	<input checked="" type="checkbox"/>
info	ntext	<input checked="" type="checkbox"/>
modified	datetime	<input type="checkbox"/>
ownerID	uniqueidentifier	<input checked="" type="checkbox"/>
ownedUntil	datetime	<input checked="" type="checkbox"/>
nextTimer	datetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
uidInstanceID	uniqueidentifier	<input type="checkbox"/>
completedScopeID	uniqueidentifier	<input type="checkbox"/>
state	image	<input type="checkbox"/>
modified	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

Tracking in SQL Server

- Esiste un servizio nativo di WF:
 - SqlTrackingService
- Configurare il DB in SQL Server:
 - C:\WINDOWS\WinFX\v3.0\Windows Workflow Foundation\SQL\EN\
 - CREATE DATABASE SqlPersistenceService
 - Tracking_Schema.sql
 - Tracking_Logic.sql
 - Requisiti:
 - MSDE/SQL2000
 - SQL 2005 (Express)

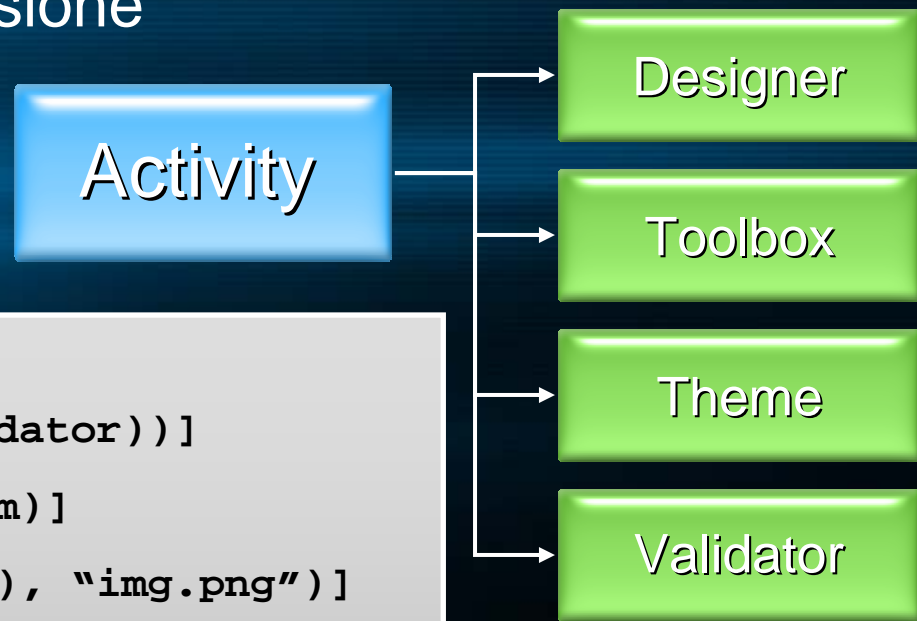


ADVANCED ACTIVITIES

010101010100010010010100101010011001
110010101010100101010101010101010101
10101010101010101111101101000010101
1010101010100010010010100101010011001
110010101010100101010101010101010101
10101010111101101000010101

Creare una Activity Custom

- Per ogni Activity custom esistono fino ad un massimo di quattro classi di “estensione”



```
[Designer(typeof(MyDesigner))]  
[ActivityValidator(typeof(MyValidator))]  
[ToolboxItem(typeof(MyToolboxItem))]  
[ToolboxBitmap(typeof(MyActivity), "img.png")]  
public class MyActivity: Activity {...}  
  
[ActivityDesignerTheme(typeof(MyTheme))]  
public class MyDesigner: ActivityDesigner {...}
```

Required
Optional

Derivare da Activity

- System.Workflow.ComponentModel.Activity
- Override di Execute
 - Restituire un ActivityExecutionStatus
- In caso di Activity transazionale da “compensare”
 - System.Workflow.ComponentModel.ICompensatableActivity

```
public enum ActivityExecutionStatus : Byte
{
    Canceling = 2,
    Closed = 3,
    Compensating = 4,
    Executing = 1,
    Faulting = 5,
    Initialized = 0
}
```

DEMO



CUSTOM ACTIVITY

010101010100010010010101010100110011
10010101010101001010101010101010101010
101010101010101011111011010000101011
10101010101000100100101001010100110011
100101010101010101010101010101010101010

DEMO



Conditioned Activity Group

01010101010001001001010101010011001
11001010101010100101010101010101010101
10101010101010101111101101000010101
1010101010100010010010100101010011001
1100101010101010101010101010101010101

Rules: l'idea

- A volte serve di condizionare l'andamento di un flusso
 - In base a regole di business dipendenti dalle entità
 - Personalizzabili a runtime/configurazione
- Quando le condizioni di base (IfElse, When, CAG) sarebbero troppo complesse
- Possiamo definire delle PolicyActivity
 - Sfruttando il designer
 - Eventualmente rendendole modificabili da un designer
- Le PolicyActivity sono dei blocchi If...Then...Else

101000100100101001010100110011
1010101001010101010101010101010101
10101010101111110110100001010111
101000100100100101010100110011
1010101001010101010101010101010101
10101010111110110100001010111

Criteri di applicazione/esecuzione

- Le regole sono applicate in base alla Priority
- In base all'esito della Rule eseguiamo Then o Else
- In caso di update di un field che modifica una delle Rule precedenti, la Rule modificata viene di solito riapplicata
 - Si parla di Dipendenza tra Rule

Criteri di dipendenza delle Policy

- Criteri di dipendenza:
 - Implicita: calcolata da WF in base alle referenze ai field
 - Attribute-based: dichiaro con specifici attributi la dipendenza
 - RuleRead
 - RuleWrite
 - RuleInvoke
 - Explicit: invocando da codice il metodo Update che aggiorna la dipendenza

Changing e Reevaluation Behavior

- Chaining behavior:
 - Full Chaining (*): è la politica di default
 - Explicit Chaining: richiede l'uso di Update(...)
 - Sequential: valutazione singola, in ordine di precedenza.
- Reevaluation behavior:
 - Always (*): il default, riapplica le regole a ritroso
 - Never: non applica alcuna regola a ritroso
- Funzioni:
 - Halt: sospende la policy

- (*) Default



DESIGNER HOSTING

010101010100010010010100101010011001
110010101010100101010101010101010101
10101010101010101111101101000010101
1010101010100010010010100101010011001
110010101010100101010101010101010101

Ospitare il designer

- Possiamo ospitare il Designer di WF nelle nostre applicazioni
- Consente ai nostri utenti di “ridisegnare” i flussi
- Come risultato possiamo avere
 - File XOML da compilare
 - Assembly .NET (come compilazione diretta del file XOML)
- Enormi potenzialità ...

Office Workflow

- Facilita gli human process
 - Lega Business Logic a item e documenti in SharePoint
 - Fornisce Tracking e contesto di esecuzione
- Fornisce agli utenti di Office Server soluzioni pronte per supportare vari scenari senza coinvolgere il reparto IT
- Fornisce soluzioni sofisticate basate su Windows Workflow Foundation (WF) integrandole con SharePoint e Office

Form

- Usate per:
 - Configurare workflow
 - Avviare workflow
 - Completare task
- ASPX Form (SharePoint e Office server)
 - Solo Server
- InfoPath Form (Office server)
 - Rendering a cura di Office Server
 - Integrazione con i client Office
 - Write once: girano su server e client

SharePoint Designer

- Scenari: scrivere app SharePoint
 - Document Workflow, es. Flusso di Review
 - Web 'Processi Guidati', es. Nuovo impiegato
 - Custom Form Action, es. su un change manda una mail
- IDE include
 - Workflow Designer
 - Basato su Wizard (Rule Wizard)
 - Building block
 - Document workflows
 - Send mail - Update List Item
 - Workflow e building block custom
 - Verifica e deployment

Office Workflow Host

- WSS fornisce
 - Persistence
 - Sul DB di Sharepoint per SPListItem
 - Event Delivery
 - Timer
 - History e Reporting
 - WSS Application Integration
- WF è ospitato nel processo WSS su tutte le macchine front-end
- Le istanze vengono deidratate nel DB

Office Activity: Esempi

- Workflow
 - Workflow History Entry
 - Audit Entry
- Task Management
 - Create Task
 - Complete Task
 - On Task Change
 - Delete Task
- Canned Tasks
 - Review Task
 - Approval Task
 - Client side scrub task
- Item
 - Update Item
 - Copy/Delete Item
 - On Item Change
- Document
 - Send To Official File
 - Get Range (XLServer)
 - Translate Document
- Web
 - Send email
 - Create alert
 - Validate User Rights
 - Request Permissions
- FP Only
 - Calculator Action
 - Create Random Number
 - Render ASPX Page to HTML
 - Assign Survey Task

Ulteriori approfondimenti

- Siti Web a proposito di WCF:
 - <http://msdn.microsoft.com/webservices/>
 - <http://www.netfx3.com/>
 - <http://wf.netfx3.com/>
- DevLeap:
 - <http://www.devleap.com/>
 - <http://blogs.devleap.com/>

Microsoft®

Your potential. Our passion.™



WinFX