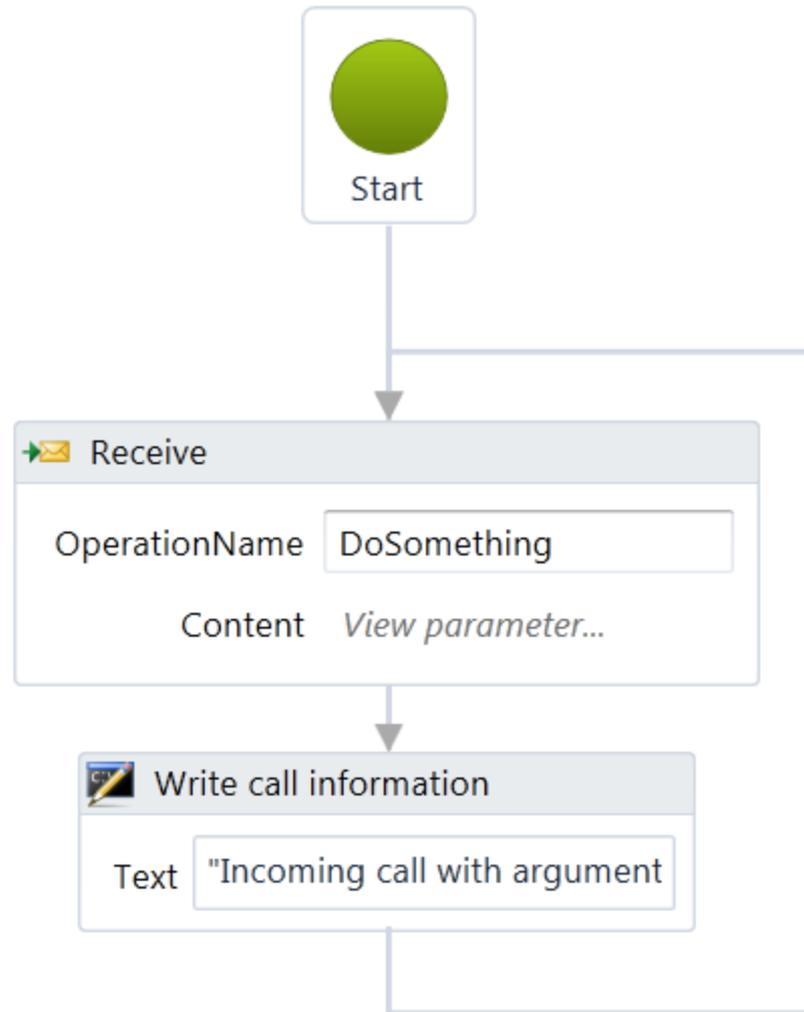


WF4 – Windows Workflow Foundation



Name: Roland König
E-Mail: roland.koenig@rkoenig.eu
Homepage: www.rkoenig.eu

Schwerpunkte: .Net- und SAP-Entwicklung

Arbeitgeber: IGZ Logistics + IT, Falkenberg



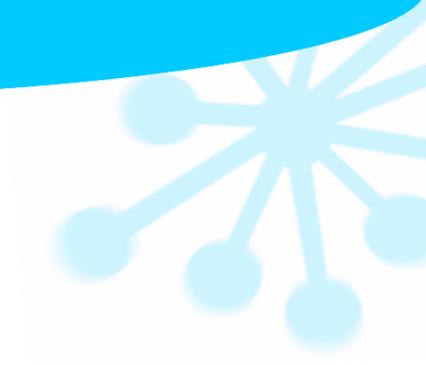


1. Allgemeines und Einsatzszenarien
2. Ein einfacher Workflow
3. Konzepte der WF4
4. Rückblick auf erstes Beispiel



WF4 – Allgemeines und Einsatzszenarien

Was ist WF?



- Was ist eigentlich mit WF3?
 - Wird nicht mehr weiterentwickelt
 - Völlig andere Code-Basis
- Was wurde durch WF4 verbessert?
 - Performance
 - Verwendung von Xaml
 - Performance des Designers
 -



WF is also *NOT*...

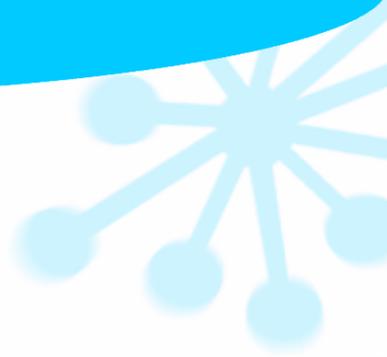


- ...a wrestling federation
- ...an international effort to protect endangered species and their habitats



Please avoid referring to WF as WWF!

Quelle: Blog von Erwyn Van Der Meer



- Laut Paul Andrew (Senior Product Manager bei Microsoft)
 - Building a Process Server (or a BPM Server)
 - Long Running Business Logic
 - Regularly Changing Business Logic or Rules
 - Require Visibility into Business Logic Execution or Model

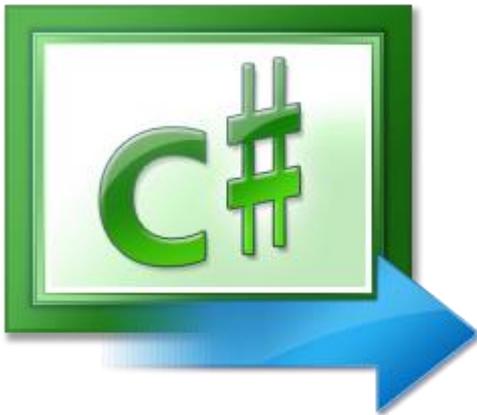
- Laut Msdn
 - SharePoint 2007 and Workflow (WF 3)
 - Human Workflow
 - Workflow and WCF Services
 - Coordinating Presentation Flow
 - Workflow Designer Re-Hosting



■ Build-Automatisierung in TFS 2010

The screenshot shows a workflow editor window titled "For Each Project in BuildSettings.ProjectsToBuild". The workflow is configured as follows:

- Foreach:** serverBl in BuildSettings.ProjectsToBu
- Body:**
 - Try to Compile the Project** (Activity)
 - Try** (Container)
 - Compile the Project** (Activity)
 - Convert Server Path to Local** (Activity)
 - Run MSBuild for Project** (Activity)
 - Catches**
 - Exception *Handle Exception*
 - Add new catch*
 - Finally** *Add an activity*



■ Beispiel GuiCustomization

Mal was ganz anderes: Workflows als Möglichkeit, Oberflächen zu Scripten. Die Darstellung jeder Zelle in einem GridView durch einen Workflow abhängig von den Werten in der Datenquelle steuern. Aktualisierung bei jeder Änderung.

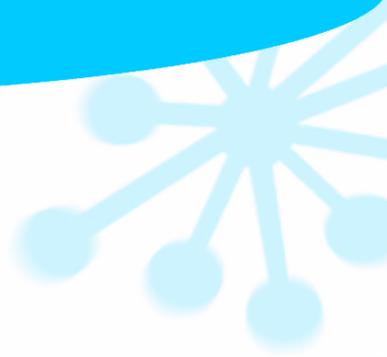
Eine gute Idee?

1. Allgemeines und Einsatzszenarien
2. Ein einfacher Workflow
3. Konzepte der WF4
4. Anwendungsfälle

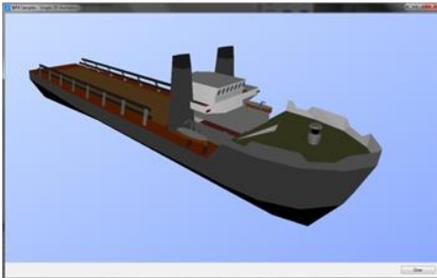


WF4 – Ein einfacher Workflow

Das Beispielprogramm



- Steuerung einer 3D-Animation
- Workflow steuert..
 - Erstellung der Objekte
 - Starten von Animationssequenzen
 - Reaktion auf Ereignisse
- Verwendet werden..
 - Standard-Activities (z. B. Delay)
 - Eigene Activities (bezogen auf 3D)



1. Allgemeines und Einsatzszenarien
2. Ein einfacher Workflow
3. Konzepte der WF4
4. Anwendungsfälle



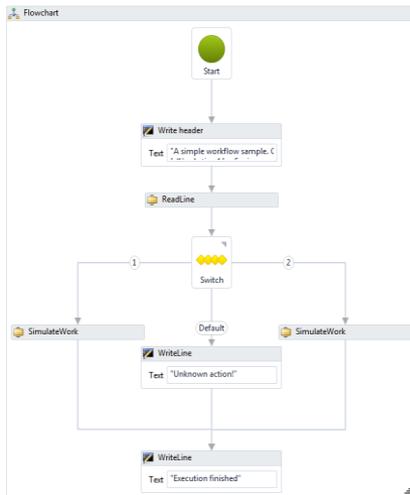
- Activity = Basisklasse aller Aktionen
- Jeder Workflow ist auch eine Activity
- Standard-Activities

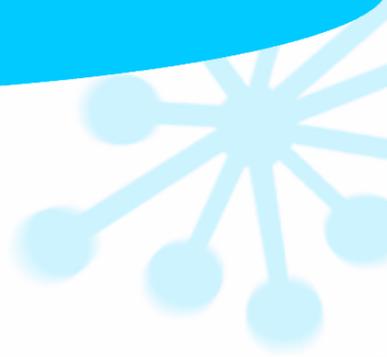
- Flowchart
- Sequence
- Send, Receive
- If, While, Do
- ...

- Eigene Activities mit C# oder anderen Sprachen

- Jede Activity hat...

- Eigene Variablen
- Eingabe- und Ausgabeparameter





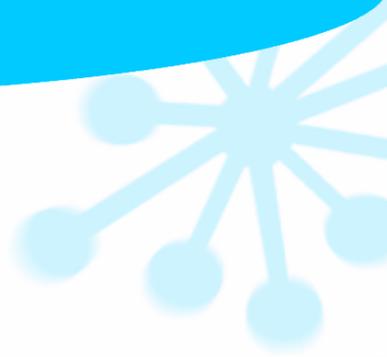
- Eigene Activities können..
 - .. mit C#-Code erzeugt werden
 - .. als Workflow erzeugt werden
- Verschiedene Basisklassen verfügbar
 - CodeActivity
 - AsyncCodeActivity
 - NativeActivity
 - ...
- Darstellung im Designer kann angepasst werden
 - Anpassung über Xaml-Dateien
 - Eigene Controls möglich
 - Technologie dafür ist WPF





■ **Beispiel CustomActivity**

Erstellung einer eigenen Activity und Anpassung der Darstellung im WorkflowDesigner

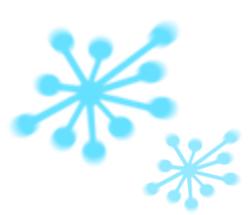


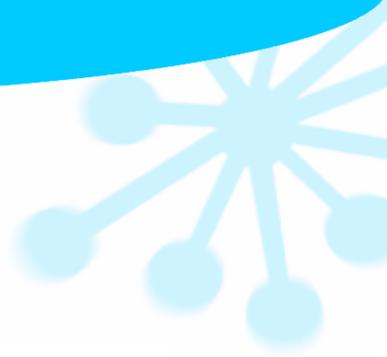
■ Hosting mit WorkflowInvoker

```
public static void Main(string[] args)
{
    MainActivity mainActivity = new MainActivity();
    WorkflowInvoker.Invoke(mainActivity);
}
```

■ Eigenschaften

- Einfach verwendbar
- Mehrmalige Aufrufe möglich
- Erster Aufruf einer Activity dauert länger
- WorkflowInvoker-Objekt kann mehrfach verwendet werden





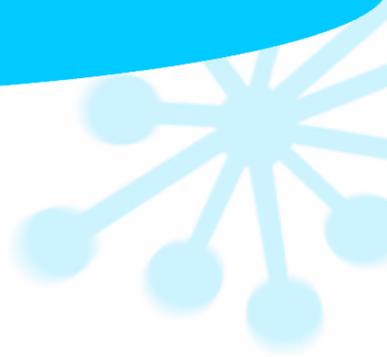
■ Hosting mit WorkflowApplication

```
MainActivity mainActivity = new MainActivity();  
WorkflowApplication wfApplication = new WorkflowApplication(mainActivity);  
wfApplication.Run();
```

■ Eigenschaften

- Asynchrone Aufrufe
- Komplexere Handhabung
- Erster Aufruf einer Activity dauert länger





■ Hosten mit WorkflowServiceHost

```
string baseAddress = "http://localhost:5020/CustomService";  
serviceHost = new WorkflowServiceHost(activity, new Uri(baseAddress));  
serviceHost.Description.Behaviors.Add(new ServiceMetadataBehavior()  
{  
    HttpGetEnabled = true  
});  
serviceHost.Open();
```

■ Eigenschaften

- Workflows als Dienste über WCF bereitstellen
- Schnittstelle im Workflow definieren
- Automatische Bereitstellung von wsdl-Dateien





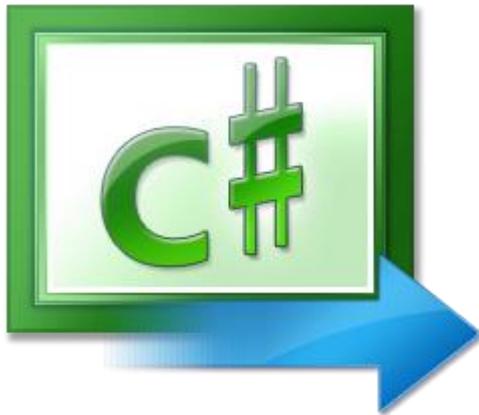
■ Beispiel WorkflowHosting

Hosten von Workflows über WorkflowInvoker und WorkflowApplication



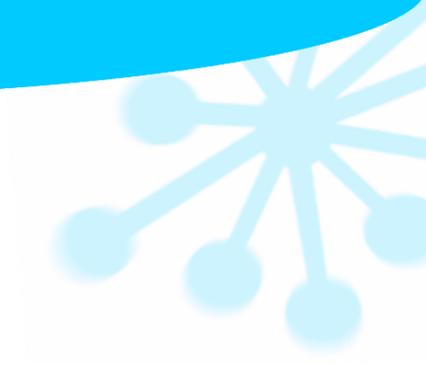
- Primitive Thread-Methoden wie z. B. Wait, Join nicht vorhanden
- Threadingverhalten abhängig von Host
 - WorkflowInvoker
 - Benutzt standardmäßig SynchronizationContext
 - WorkflowApplication
 - Benutzt standardmäßig ThreadPool
 - SynchronizationContext kann gesetzt werden

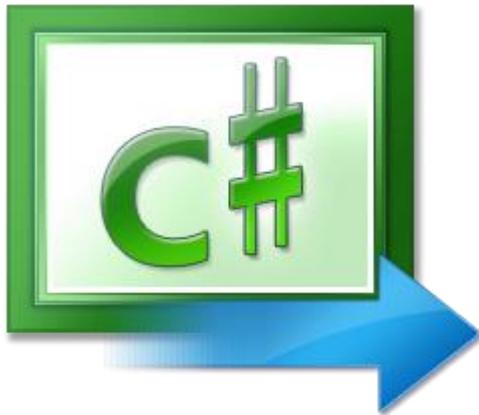




■ Beispiel `ActivityAndThreading`

Hosten von Workflows über `WorkflowInvoker` und `WorkflowApplication`



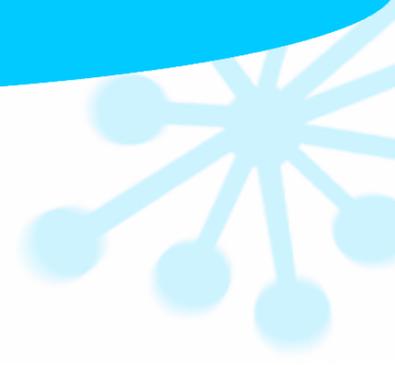


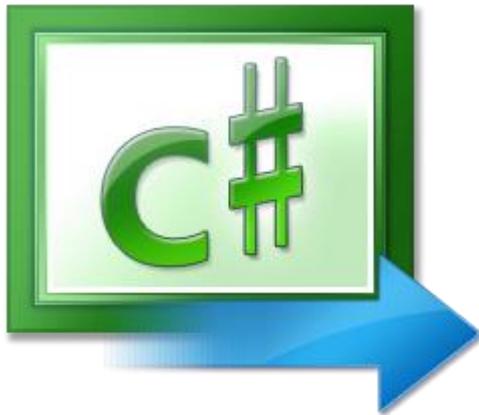
■ Beispiel BackgroundWorkerActivity



WF4 – Konzepte

Workflow Services

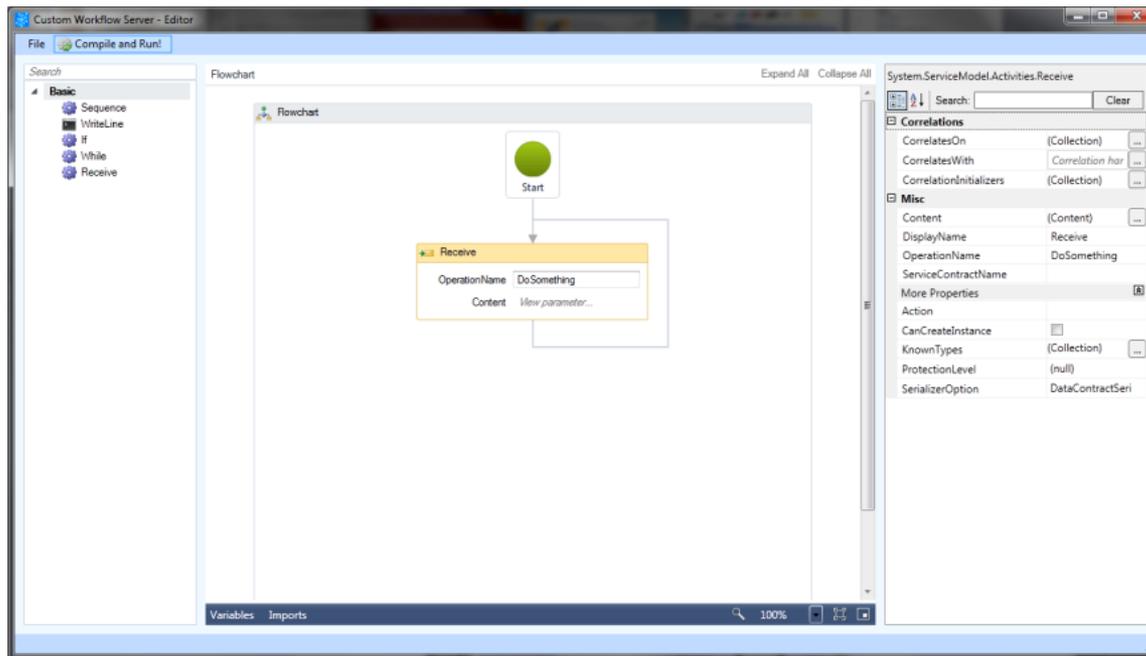




■ Beispiel CustomWorkflowService

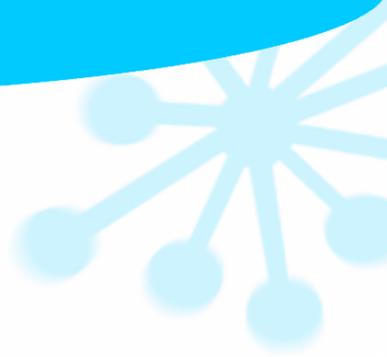
Einen Workflow per Designer definieren und direkt als Webservice starten

- Workflowdesigner in eigenen Programmen verwenden
- Erste Ergebnisse schnell sichtbar
- Detail-Anpassungen teils schwierig



WF4 – Agenda

1. Allgemeines und Einsatzszenarien
2. Ein einfacher Workflow
3. Konzepte der WF4
4. Anwendungsfälle





■ Quellen:

- Paul Alans Block

<http://blogs.msdn.com/b/pandrew/archive/2007/02/01/what-to-use-windows-workflow-foundation-for.aspx>

- Msdn

<http://msdn.microsoft.com/en-us/library/cc709458.aspx>

