# Requirements

Visual Studio 2010 as Administrator

Machine needs AppFabric installed and configured

ZoomIT!

# Demo 1 - Hello World in Workflow Foundation 4

New 🡺 Activity Library

Add the following arguments:

* Name: ClientsName, Direction: In, Type: String
* Name: Balance, Direction: Out, Type: Decimal (will need to browse for Decimal)

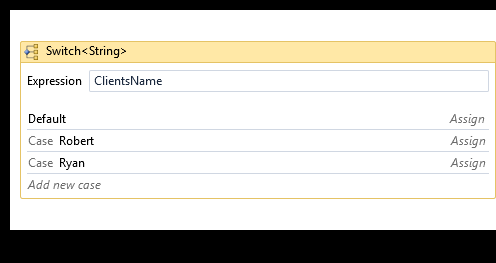
Add Switch component to form 🡺 type is string 🡺 Expression is ClientsName.

Add a case for Robert and Ryan.

Click on Default case 🡺 Add assign component 🡺 Balance = 0

Click on “Robert” case 🡺 Add assign component 🡺 Balance = -123.234D

Click on “Ryan” case 🡺 Add assign component 🡺 Balance = 5563



Add new project 🡺 Console Application

Add reference to

* activity library created above
* System.Activities

Add code to Main method

1. string clientName = "Robert";
2. Activity1 activity = new Activity1();
3. activity.ClientsName = clientName;
4. IDictionary<string,object> results = WorkflowInvoker.Invoke(activity);
5. Console.WriteLine("{0} has {1:C}", clientName, results["Balance"]);

Demo! Try replacing names and showing results.

Remove all code in main and replace with:

1. Activity wf = new WriteLine()
2. {  
    Text = "Hello World."
3. };
4. WorkflowInvoker.Invoke(wf);

# Demo 2 – Declarative Workflow Services

Add new Workflow Service Application

Click Variables and delete data variable and add the following:

* Name: ClientsName, Type: String
* Name: Balance, Type: Decimal

Change the RecieveRequest OperationName to CheckBalance

Click the Content 🡺 Set the Message Data to: ClientsName and type to String

Click the SendResponse Content 🡺 Set the Message Data to: Balance and Type to Decimal

Copy the Switch from previous activity library

Demo!

Show in WCF client how it works

When done show the Web.Config is clean

Right click on service project 🡺 Publish 🡺

* Method: Web Deploy
* Service URL: localhost
* Site: Default Web Site/Demo
* Mark as application: Ticked

# Demo 3 – Tour of AppFabric

Start up IIS

Show the three new buttons on ROOT, Site, Virtual Directory.

In Virtual Directory, show services, then end points.

Add new console application 🡺 Add service reference to: http://localhost/Demo/Service1.xamlx

In Main add

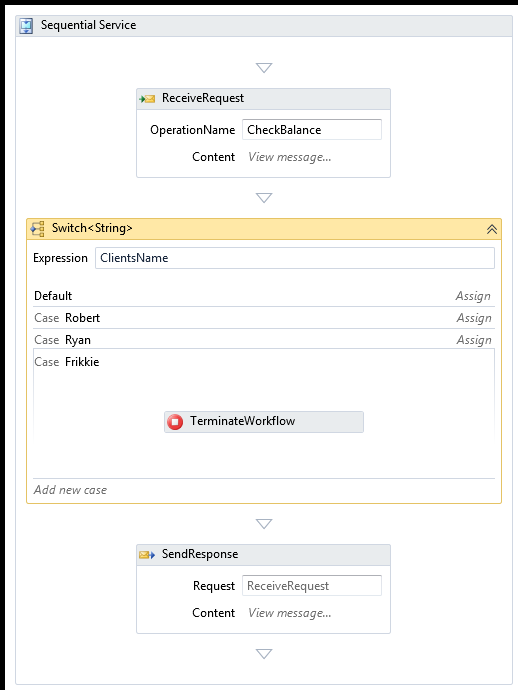
1. IService service = new ServiceClient();
2. Console.WriteLine(service.CheckBalance(new CheckBalanceRequest("Ryan"))[.@decimal.Value](mailto:.@decimal.Value));

Demo

Switch back to app fabric dash and show WCF calls and then WF instances – show right click on WF instances

Now jump back to the Declarative service and add a third case “Frikkie” 🡺 add a TerminateWorkflow activity

* Exception: New Exception()
* Reason: “blah blah”



Rebuild and republish

Change console app 2 to call with Frikkie and then have it fail.

Show dashboard for fail (may take a moment to run)

# Demo 4 – Death of a Windows Service

Not actually doing it, just talk about it.

AppFabric 🡺 Configure 🡺 Auto-Start 🡺 Enabled

Explain app pool change

Jump back to Declarative Service 🡺 Add App\_Code 🡺 Add AppStart.cs class 🡺 Add following code:

1. public static void AppInitialize()
2. {
3. }