FlowTasks – Installation Guide

Contents

[Dependencies 1](#_Toc352014620)

[Installing AppFabric 1](#_Toc352014621)

[Database 2](#_Toc352014622)

[Building FlowTasks 2](#_Toc352014623)

[Installation 2](#_Toc352014624)

[Workflows 3](#_Toc352014625)

[Connectors 5](#_Toc352014626)

[DocsOnFolder 5](#_Toc352014627)

[Twitter 6](#_Toc352014628)

[Web 6](#_Toc352014629)

[Task List 7](#_Toc352014630)

[Dashboard 9](#_Toc352014631)

[Report 9](#_Toc352014632)

[Sketch 10](#_Toc352014633)

[Conclusion 11](#_Toc352014634)

# Dependencies

Make sure you have the following installed in your PC:

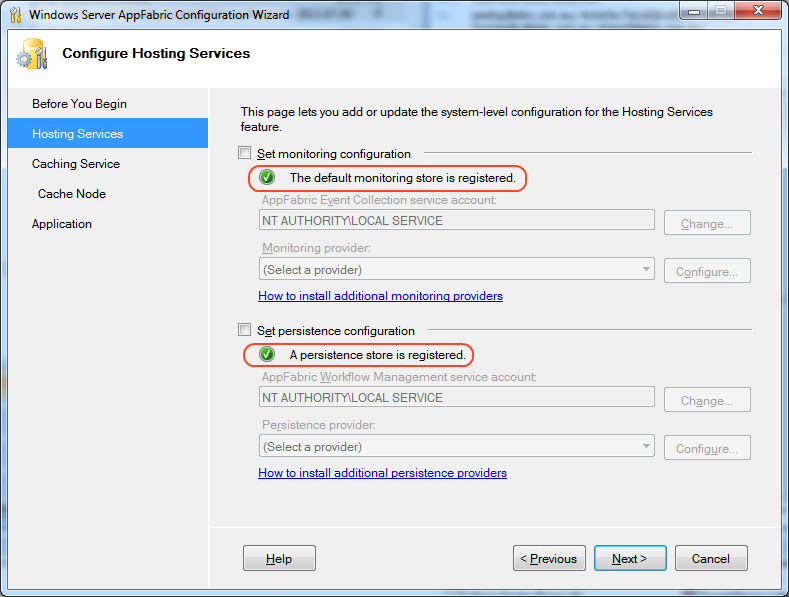
Windows Server AppFabric

SQL server 2008 or higher

Visual Studio 2012

# Installing AppFabric

After you install AppFabric , configure the monitor and persistence sections.



# Database

Under the SQL folder you will find the scripts to create the databases that the framework needs.

# Building FlowTasks

After you download the code build all the projects and run the tests to make sure that everything is all right.

**Make sure you check all the configuration files and change them according to your environment settings.**

**Open the solution FlowTasks.sln in Visual Studio 2012 and build all the projects. You can also build all the projects using the build script**

****

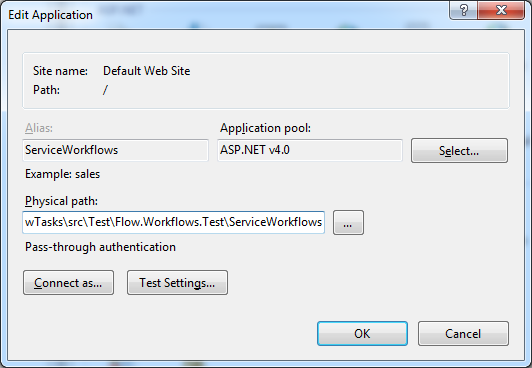
**Which it uses powershell script and the psake [**<https://github.com/psake/psake>]

# Installation

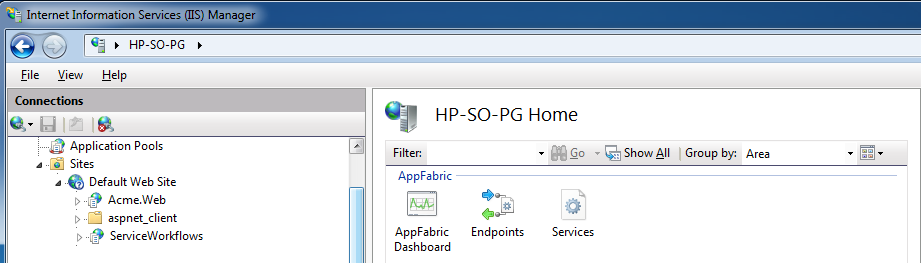
To install FlowTasks follow the following steps.

## Workflows

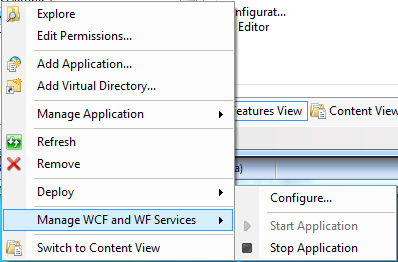
Open IIS manager, create “ServiceWorkflows” application if doesn’t already exists.



Physical path should be *Test\Flow.Workflows.Test\ServiceWorkflows.*

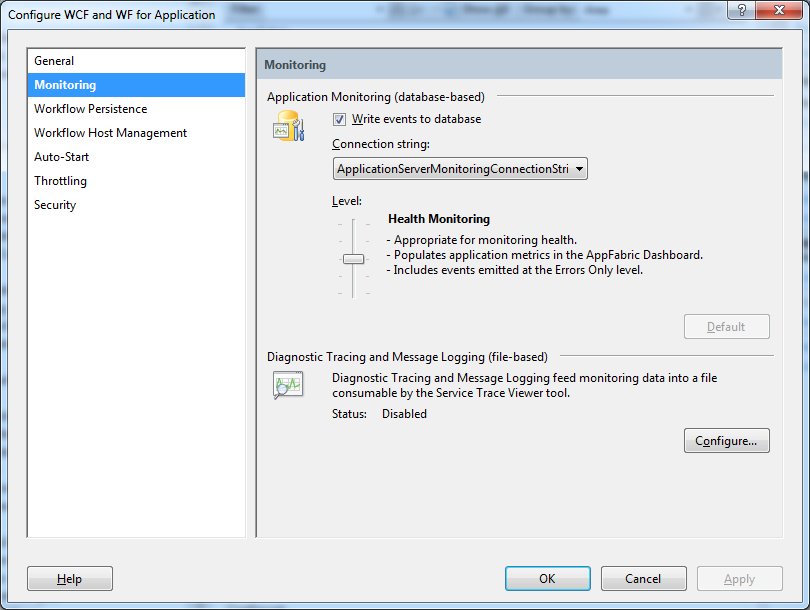


If you right click on ServiceWorkflows,

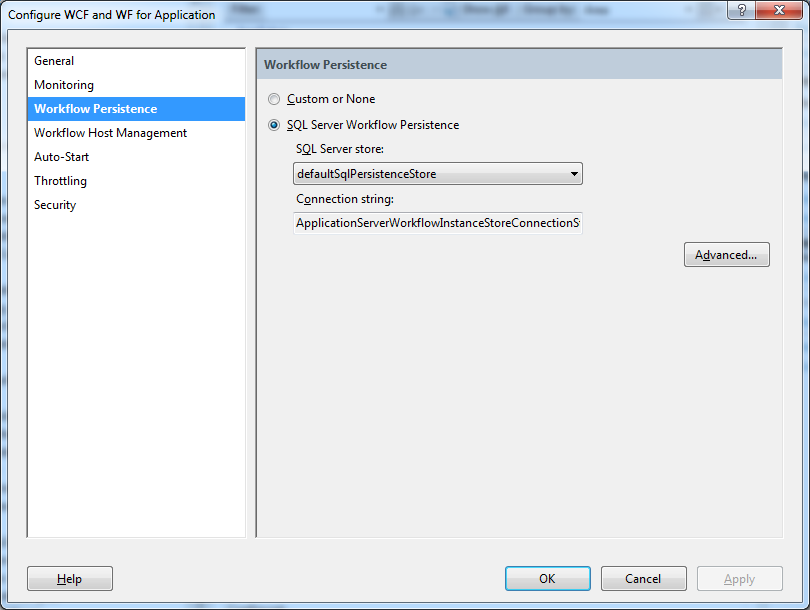


Make sure the application is started; then select Configure...

Make sure Monitoring have write events to database selected,

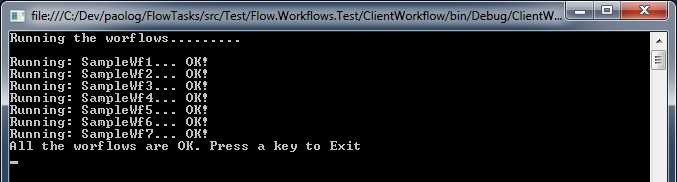


And Workflow Persistence has SQL Server Workflow Persistence selected



To test the workflows run the ClientWorkflow project. This is a console application that runs the workflows.

Before you can continue you need to pass this test.



## Connectors

Under Flow.Connectors you will find two examples of connectors. A connector is an entity that triggers an event that starts a workflow.

### DocsOnFolder

This is a windows service that monitors a folder. Every time someone drops a file in the folder a workflow will start.

To install the windows service use the command: installutil <yourpath>DocsOnFolderService.exe

### Twitter

This is a windows service that monitors a twitter user, every time someone sends a tweet with a specified pattern a workflow will start.

To install the windows service use the command: installutil <yourpath> Flow.Connectors.Twitter.exe

## Web

Flow.Tasks.Web project can be used to manage the tasks assigned the a user; it has a Dashboard that an Admin user can use to monitor existing or completed workflows; a Report system that can be used to fine-tuned the business processes; a workflow designer that business analysts can use to communicate the solution to the development team.

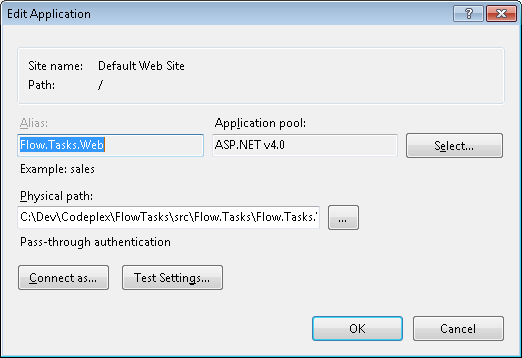
The table below explain what application a role can access.

|  |  |
| --- | --- |
| **Application** | **Role** |
| Task List | All roles |
| Dashboard | Admin |
| Report | Project Manager |
| Sketch | Business Analyst |

If you are using the scripts provided there are already some sample users that you can sue for testing.

|  |  |
| --- | --- |
| User | Role |
| CGRANT | Developer, Admin, Project Manager |
| PNEWMAN | DEVELOPER |
| HBOGART | Business Analyst |

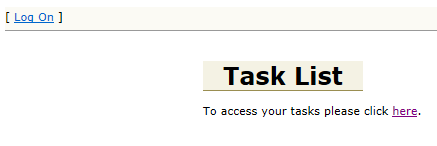
Open IIS manager, create “Flow.Tasks.Web” application if doesn’t already exists.



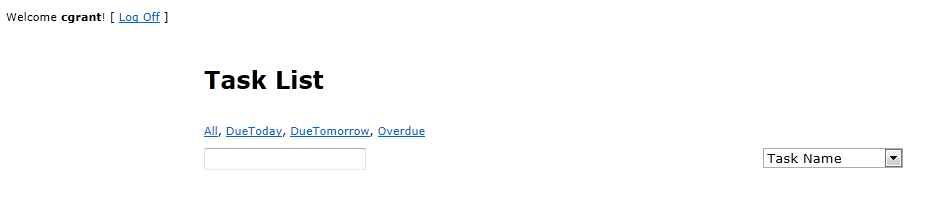
Physical path should be *Web\Flow.Tasks.Web.*

### Task List

In the task list a user can find all the tasks assigned to him/her.



After you login, if you used the script provided to create the users database you will have some test users, using for example <cgrant, pwd> you can now access the task list, which at this point should be empty because there aren’t any active workflow yet.



To start a workflow you can use the connectors or a quicker way is to use the ClientWorkflow project.

In ClientWorkflow change the code in Program.cs like this:

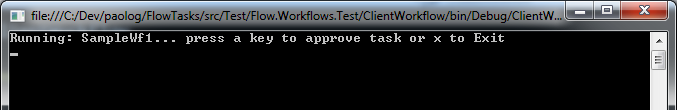
static void Main(string[] args)

{

ServiceCallsWf1(false);

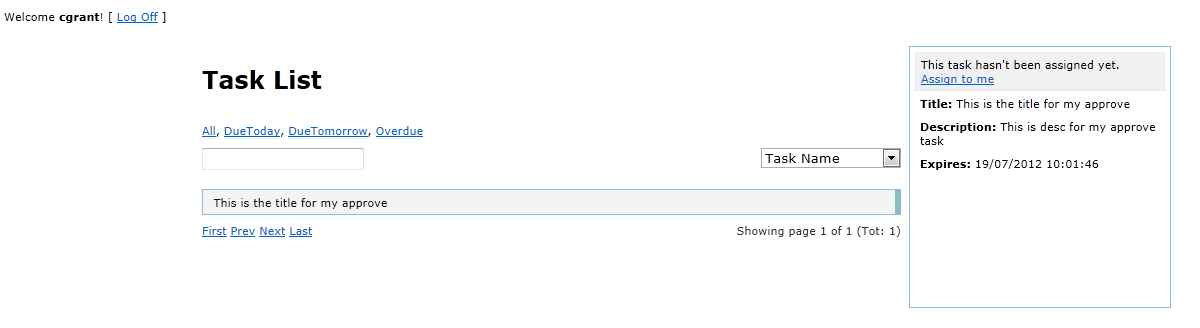
}

And now run the project.

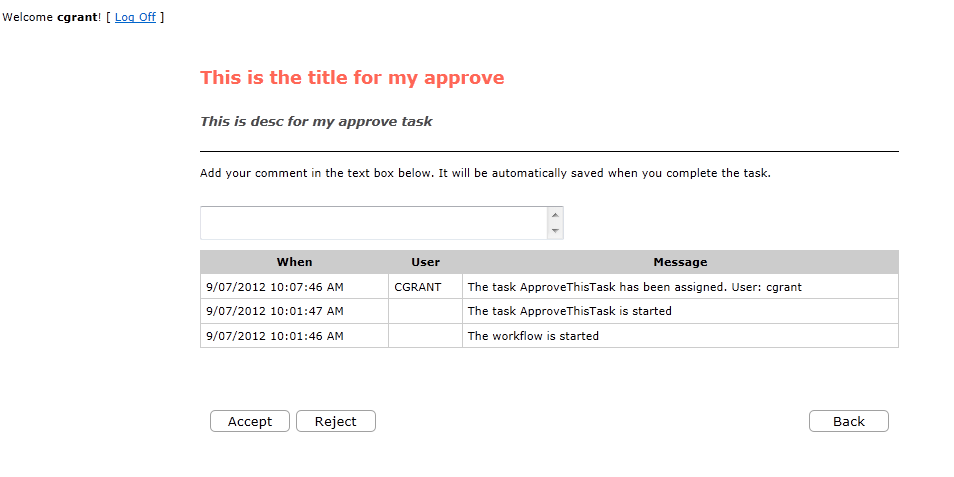


Press “x” and exit the console application.

Now starts again Flow.Tasks.Web and login <cgrant, pwd>



And here you see the task that has been assigned to you. If you click on the task, you’ll see the Task View where you can complete it.

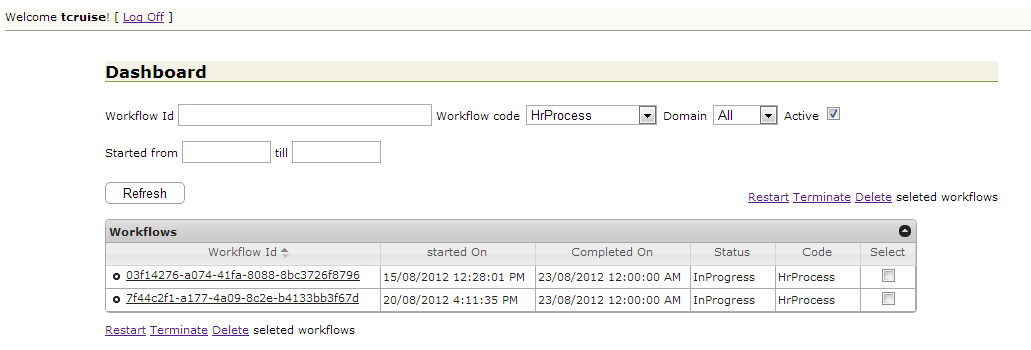


### Dashboard

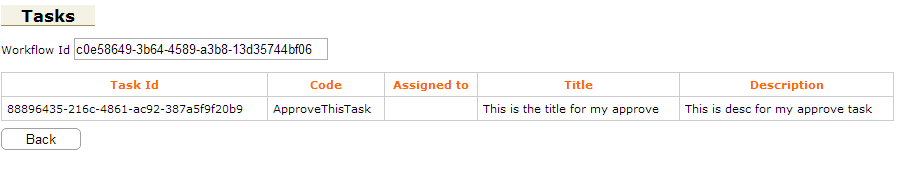
The dashboard is an application used to manage all the workflows in the system.

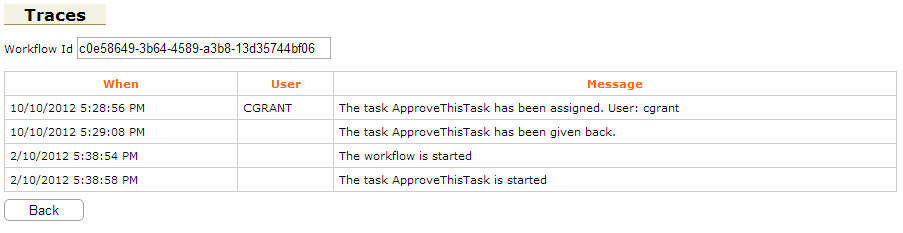
You can terminate existing workflows or restart some others or check the history for a particular workflow or check what the active tasks in a workflow are.

You need to login as an admin user to access this functionalities <cgrant, pwd>.



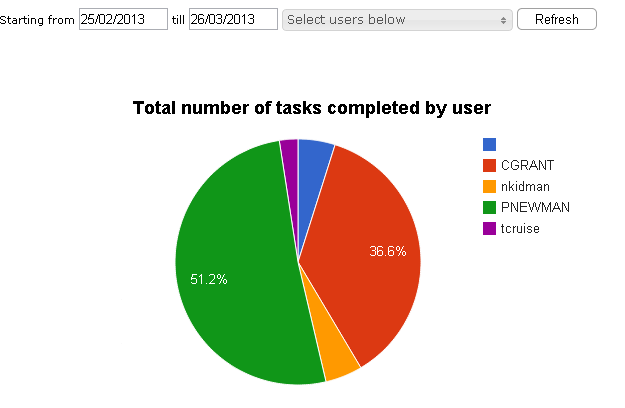
If you click on a workflow id, you can then access the tasks and traces.

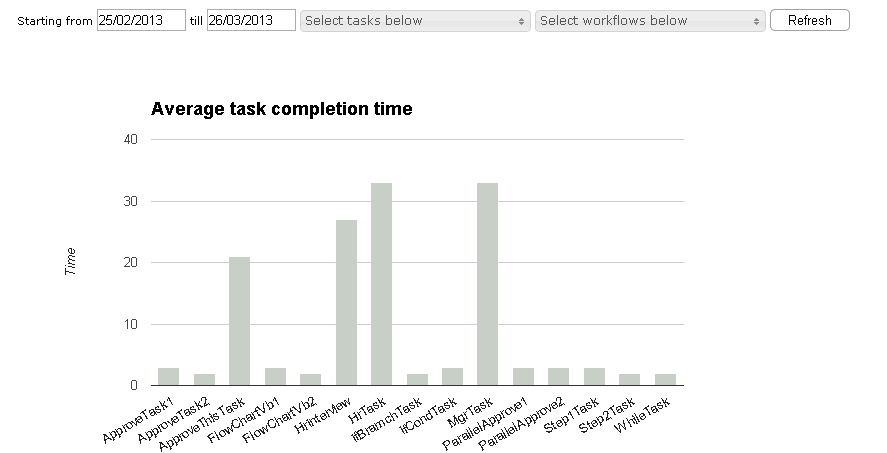




### Report

FlowTasks collects a lot of information while processing the workflows. All this information can be viewed through some reports. As an example:

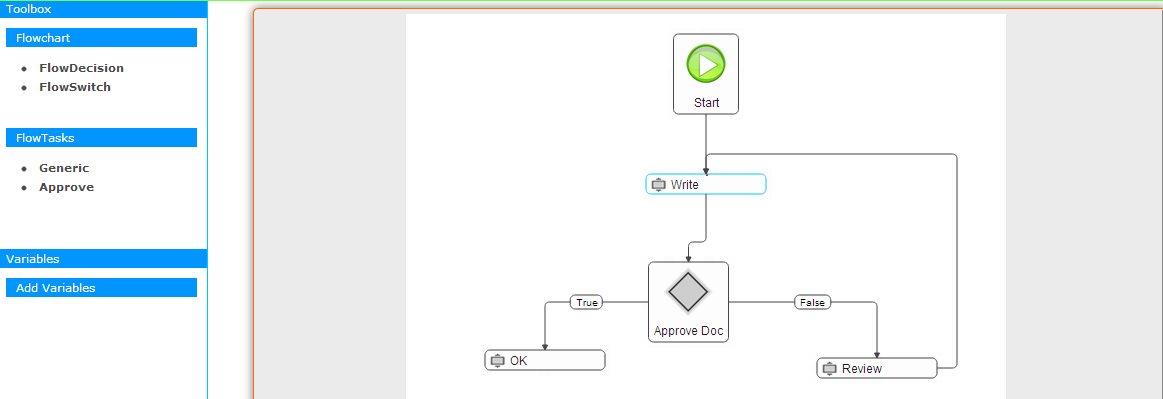




You need to be a Project Manager <cgrant, pwd> to access the Report.

### Sketch

With Sketch everyone can easily create a workflow, without be a developer. All levels of the business can use Sketch to design the process workflow. This eliminates the communication gap that sometimes exists between developers and other parts of the business.



You need to be a Business Analyst to access Sketch <hbogart, pwd>.

The developers can then complete the development and deploy the workflow. The workflow saved from Sketch can be directly loaded into Visual Studio 2012.

# Conclusion

At this point FlowTasks has been installed and you can now start to create your own workflows.