Orleans SDK – Setup and Quick Start

For the Orleans v0.8.0 Alpha release, April-2012

This Quick-Start guide is intended to get you up and running with an Orleans development environment in the shortest amount of time. Once you have completed the steps in this document, you should;

* Have a working Orleans silo on your local machine
* Be able to compile and run the sample applications
* Have a Visual Studio Development environment prepared for writing Orleans grains and clients
* Understand which documentation to turn to for next steps in using Orleans.

[1 Introduction 2](#_Toc323287958)

[1.1 What is Orleans? 2](#_Toc323287959)

[1.2 What is Horton? 2](#_Toc323287960)

[1.3 License 2](#_Toc323287961)

[1.4 Orleans SDK Quick Start Steps 2](#_Toc323287962)

[2 Orleans Setup for Developers 3](#_Toc323287963)

[2.1 Installing the Orleans SDK 3](#_Toc323287964)

[2.2 Install the Orleans project templates into Visual Studio 2010. 3](#_Toc323287965)

[2.3 Running the Local Orleans Development Silo 3](#_Toc323287966)

[3 Orleans SDK - Sample Applications 4](#_Toc323287967)

[3.1 Hello Orleans 4](#_Toc323287968)

[3.2 Chirper 4](#_Toc323287969)

[3.3 Orleans Azure Web Sample 5](#_Toc323287970)

[3.4 Horton Graph Samples 6](#_Toc323287971)

[4 Writing Your Own Orleans Applications 6](#_Toc323287972)

[4.1 Developing Your Own Orleans Application 6](#_Toc323287973)

[4.2 Deployment of an application to the Local Orleans Development Silo 6](#_Toc323287974)

[4.3 Deployment to Remote Machines Using Orleans Deployment PowerShell Scripts 7](#_Toc323287975)

[5 Orleans Documentation 7](#_Toc323287976)

[5.1 Orleans Quick Start Guide 7](#_Toc323287977)

[5.2 Orleans Programmer’s Guide 7](#_Toc323287978)

[5.3 Orleans Configuration Guide 7](#_Toc323287979)

[5.4 Running Orleans on Windows Azure 7](#_Toc323287980)

[5.5 Orleans Deployment Scripts and Host Pre-requisite Guide 8](#_Toc323287981)

[5.6 Orleans API Reference 8](#_Toc323287982)

[5.7 Horton API Reference 8](#_Toc323287983)

[5.8 Orleans SOCC 2011 publication 8](#_Toc323287984)

[5.9 Technical Report – Horton Graph Library 8](#_Toc323287985)

[6 More Information 9](#_Toc323287986)

[6.1 Public (external) information: 9](#_Toc323287987)

[6.2 Microsoft Internal Information and Feedback 9](#_Toc323287988)

# Introduction

The Orleans SDK contains the Orleans runtime system, libraries for application development, Horton Graph Database, and samples showing how to build applications using Orleans and Horton.

This document provides a quick start intro to the contents of the Orleans SDK, and is designed to walk you through your first 15 minutes exploring the Orleans functionality.

See the Documentation and More Information sections below for pointers to more details on how to develop and deploy Orleans applications.

## What is Orleans?

Orleans is a framework for building highly-scalable cloud applications. Its programming model and distributed runtime system assume much of the burden involved in developing, deploying, and operating a highly scalable application running on potentially unreliable servers. Orleans provides a higher level programming model than existing .NET libraries, significantly reducing the amount of work that developers need to do while easing the transition from desktop or client/server systems to cloud-scale distributed applications. In Orleans, applications are created as a composition of interacting actors called grains that are hosted within large, usually one per machine, container processes called silos.

## What is Horton?

Horton is a distributed graph database designed for efficient handling of very large graph data structures. It provides interfaces for defining the data as graph entities, such as nodes and edges, uploading the data into a cluster of machines, retrieving the uploaded data and executing reachability queries against the data. Horton is written as an Orleans application.

## License

The Orleans SDK Alpha release is **Microsoft Confidential**, and for internal use only.

## Orleans SDK Quick Start Steps

Quick-start steps in a nutshell:

1. Unzip the SDK drop
2. Install Orleans project templates into Visual Studio
3. Start the Orleans local development silo, which has several sample applications pre-installed
4. Try some of the sample application client applications in the SDK

Longer term:

1. Read Orleans Programmer’s Guide
2. Write your own Orleans application(s)
3. Test your own Orleans application(s) using the local development silo
4. Prepare remote machines for acting as Orleans hosts
5. Deploy, manage and monitor Orleans on remote machine clusters using the Orleans PowerShell scripts
6. Package your Orleans application for deployment to Azure, deploy and test the application running in Azure

# Orleans Setup for Developers

## Installing the Orleans SDK

The Orleans SDK is shipped as a single zip file that should be unzipped on your the local hard drive.

The recommended installation location for the Orleans SDK is C:\Orleans-SDK. We will refer to the actual directory where the SDK contents are located as [ORLEANS-SDK] throughout the rest of this document.

## Install the Orleans project templates into Visual Studio 2010.

The Orleans SDK contains a VSIX package with Visual Studio project and item templates for use with Orleans.

If you previously installed the Orleans project templates, you need to uninstall them by running the[ORLEANS-SDK]\ UninstallOrleansVSTools.cmd script file.

To install the project templates, run the [ORLEANS-SDK]\InstallOrleansVSTools.cmd script file to install the templates into Visual Studio 2010 and set up the required environment variable settings.

Naturally, this step is only required on developer machines that will be used for developing Orleans applications, and is not required on Orleans runtime host machines.

To later uninstall the Visual Studio templates for Orleans, just run the [ORLEANS-SDK]\UninstallOrleansVSTools.cmd script file.

## Running the Local Orleans Development Silo

The Orleans SDK includes a local development silo which is pre-configured to run on localhost.

This local silo will allow initial development and testing of Orleans samples and applications, and can also be used as an initial verification step before deployment of changes into a larger cluster of machines.

The Orleans local development silo can be started in a console window using the command script:  
[ORLEANS-SDK]\**StartLocalSilo.cmd**

Once started, the silo can be stopped using Ctrl-C or closing the console window.

For the current Alpha release, an Orleans silo must be stopped and restarted in order to pick up any application changes.

# Orleans SDK - Sample Applications

Several sample applications are included with the Orleans SDK.

Providing the Orleans local development silo is already running (see section 2.3) you should be able to open the solution file for each sample application in Visual Studio, and then build and run out of the box simply by pressing ‘F5’.

## Hello Orleans

This is a very simple application that just implements the classic “Hello World” program using an Orleans grain.

Source code location = [ORLEANS-SDK]\Samples\HelloOrleans\  
VS solution file = [ORLEANS-SDK]\Samples\HelloOrleans\HelloOrleans.sln  
Client binaries location = Build Client.ConsoleApp project in Visual Studio  
Grain binaries location = [ORLEANS-SDK]\LocalSilo\Applications\HelloOrleans\

Steps to run the sample:

1. Ensure local Orleans silo is already running
2. Open [ORLEANS-SDK]\Samples\HelloOrleans\**HelloOrleans.sln** in Visual Studio
3. Select **Client.ConsoleApp** project as the Visual Studio startup-project if it is not already
4. Press ‘F5’ to build and run Client.ConsoleApp
5. Note the message returned back from Orleans in the console window

## Chirper

This is the source code for the Chirper demo application featured at TechFest 2011 and forms the basis for much of the discussion of Orleans features in the Orleans Programmer’s Guide.

Source code location = [ORLEANS-SDK]\Samples\Chirper\  
VS solution file = [ORLEANS-SDK]\Samples\Chirper\Chirper.sln  
Client binaries location = [ORLEANS-SDK]\Binaries\ChirperClient\  
Grain binaries location = [ORLEANS-SDK]\LocalSilo\Applications\Chirper\

Steps to run the sample:

1. Ensure local Orleans silo is already running.
2. Run the [ORLEANS-SDK]\Samples\Chirper\**2.Demo Loader.cmd** script to run the Chirper loader app and create some sample data for this application.
3. Run the [ORLEANS-SDK]\Samples\Chirper\**3.Demo UI.cmd** script to run the Chirper UI client app.
4. Run the [ORLEANS-SDK]\Samples\Chirper\**4.Demo Console.cmd** script to run the Chirper console client app.
5. Log in to the Chirper UI client app using the default username prompted (no password required)
6. Type a chirp message into the status test box on the Chirper UI.
7. Note the message echoed in the UI client app windows and in the Console client app window.

## Orleans Azure Web Sample

This is an example application that shown how to run Orleans silos and a web UI client in Windows Azure.

Source code location = [ORLEANS-SDK]\Samples\AzureWebSample\  
VS solution file = [ORLEANS-SDK]\Samples\AzureWebSample\AzureWebSample.sln  
Client binaries location = Build sample in Visual Studio and run in Azure Developer Fabric[[1]](#footnote-1)  
Grain binaries location = [ORLEANS-SDK]\LocalSilo\Applications\HelloOrleans\

Steps to run the sample:

1. Ensure you have the Windows Azure Tools for Visual Studio 2010 v1.4 installed
2. Run Visual Studio 2010 with **elevated** Administrator permissions [so that Azure developer environment will work]
3. Open [ORLEANS-SDK]\Samples\AzureWebSample\**AzureWebSample.sln** in Visual Studio
4. Select **OrleansAzureSample** project as the Visual Studio startup-project if it is not already
5. Press ‘F5’ to build and run the sample in the local Azure compute emulator
6. When the web browser window opens, press the “Say Hello to Orleans” button
7. Note the message returned back from Orleans in the web browser window

See the document “Running Orleans on Windows Azure” for more background information on using Azure with Orleans.

## Horton Graph Samples

This is an example application that loads Horton Graph Library data into one or more Orleans silos.

Source code location = [ORLEANS-SDK]\Samples\GraphSamples\  
VS solution file = [ORLEANS-SDK]\Samples\GraphSamples\GraphSamples.csproj  
Client binaries location = [ORLEANS-SDK]\Binaries\GraphClient\  
Grain binaries location = [ORLEANS-SDK]\LocalSilo\Applications\GraphLibrary\

Steps to run the sample:

1. Ensure local Orleans silo is already running.
2. Open [ORLEANS-SDK]\Samples\GraphSamples\**GraphSamples.csproj** in Visual Studio
3. Select **GraphSamples** project as the Visual Studio startup-project if it is not already
4. Press ‘F5’ to build and run GraphSamples
5. Note the messages returned back from Orleans in the console window

# Writing Your Own Orleans Applications

## Developing Your Own Orleans Application

See the Orleans Programmer’s Guide below for more details on how to write Orleans applications.

## Deployment of an application to the Local Orleans Development Silo

A typical Orleans application is comprised of one DLLs containing grain interfaces and implementation classes. Only those binaries need to be deployed to the application folder of a silo.

Just copy new / updated application DLLs into the Orleans applications directory   
[ORLEANS-SDK]\LocalSilo\Applications

The recommended practice is to use a separate sub-directory for each separate application.   
For example, the HelloOrleans application is located in   
[ORLEANS-SDK]\LocalSilo\Applications\**HelloOrleans**

For the current Alpha release, an Orleans silo must be stopped and restarted in order to pick up any application changes.

## Deployment to Remote Machines Using Orleans Deployment PowerShell Scripts

Deployment of Orleans silos and applications onto individual or cluster of remote host machines silo can also be done using the **Orleans Deployment PowerShell scripts**. The scripts simply automate the job of copying of the Orleans runtime and applications to a set of machines, and remotely starting Orleans silos on them, as well as tasks, such un-deployment, starting/stopping/monitoring silos, and gathering silo logs. The same can be done by other remote deployment tools, such as Autopilot, System Center, etc. The PowerShell scripts are included only to bootstrap testing on a cluster of machines.

The PowerShell scripts use a deployment manifest file to drive the deployment process.

To run the Orleans deployment scripts, start PowerShell in Administrator mode, then run the script:  
[ORLEANS-SDK]\RemoteDeployment\DeployOrleansSilos.ps1 path-to-deployment-manifest.xml

More details on the full suite of Orleans Deployment PowerShell Scripts is provided in the following document:  
[ORLEANS-SDK]\Docs\Orleans-Remote-Deployment.docx

# Orleans Documentation

## Orleans Quick Start Guide

Information on getting started with Orleans is contained in this “Orleans SDK Quick Start” document [This document]

Document location = [ORLEANS-SDK]\Docs\Orleans-SDK-Quick-Start.docx

## Orleans Programmer’s Guide

Detailed information on writing Orleans applications is contained within the “Programmer’s Guide to Orleans” document.

Document location = [ORLEANS-SDK]\Docs\Programmer’s Guide to Orleans.docx

## Orleans Configuration Guide

Information on the key configuration parameters and how they should be set for the most common scenarios.

Document location = [ORLEANS-SDK]\Docs\Orleans-Configuration.docx

## Running Orleans on Windows Azure

Detailed information on how to host Orleans applications in Windows Azure is contained in the “Running Orleans on Windows Azure” document.

Document location = [ORLEANS-SDK]\Docs\Running Orleans on Windows Azure.docx

## Orleans Deployment Scripts and Host Pre-requisite Guide

Information on obtain and configuring the pre-requisite packages required to run Orleans on a Windows Server host machine is contained in this “Orleans Deployment” document.

Document location = [ORLEANS-SDK]\Docs\Orleans-Remote-Deployment.rtf

## Orleans API Reference

Detailed information on the Orleans application programming interface is contained within the “Orleans API Reference” document.

Document location = [ORLEANS-SDK]\Docs\Orleans-API.chm

## Horton API Reference

Detailed information on the Horton Graph Library application programming interface is contained within the “Horton Graph Library API Reference” document.

Document location = [ORLEANS-SDK]\Docs\Graph-API.chm

## Orleans SOCC 2011 publication

An overview of the Orleans system, including detailed information on the motivations and technology used under the covers by Orleans from the proceedings of the 2011 Symposium On Cloud Computing.

Document location = [ORLEANS-SDK]\Docs\Orleans-SOCC-2011.pdf

## Technical Report – Horton Graph Library

An overview of the Horton system, including investigation of the graph query execution engine and how Horton has been used to manage a large graph for a social networking application called Codebook.

Document location = [ORLEANS-SDK]\Docs\Horton-VLDB2011-Demo-Paper.pdf

# More Information

## Public (external) information:

* Orleans public home page:  
  <http://research.microsoft.com/projects/orleans>
* Horton Graph Library public home page:  
  <http://research.microsoft.com/projects/ldg/>
* Channel 9 video introduction to Orleans  
  <http://channel9.msdn.com/Shows/Going+Deep/Project-Orleans-A-Cloud-Computing-Framework>
* Orleans SOCC 2011 paper:   
  <http://research.microsoft.com/apps/pubs/?id=141999>
* News Articles about Orleans
  + **Orleans: Microsoft's next-generation programming model for the cloud (**Mary-Jo Foley)  
    <http://www.zdnet.com/blog/microsoft/orleans-microsofts-next-generation-programming-model-for-the-cloud/7152?tag=mantle_skin;content>
  + **Orleans: More on Microsoft's cloud programming model in the sky (**Mary-Jo Foley)  
    <http://www.zdnet.com/blog/microsoft/orleans-more-on-microsofts-cloud-programming-model-in-the-sky/7830>
  + **5 Futuristic Microsoft Technologies (**Mary-Jo Foley)  
    <http://redmondmag.com/articles/2011/02/01/5-futuristic-microsoft-technologies.aspx>

## Microsoft Internal Information and Feedback

* Orleans Discussion DL :- [OrlTalk@microsoft.com](mailto:OrlTalk@microsoft.com)

Join group: <http://idwebelements/GroupManagement.aspx?Group=OrlTalk&Operation=join>   
Leave group: <http://idwebelements/GroupManagement.aspx?Group=OrlTalk&Operation=leave>

* Direct feedback DL to the Orleans Development Team : [orldev@microsoft.com](mailto:Orleans@microsoft.com)
* Orleans SDK Documentation: <Orleans_Documentation.html>
* Orleans Toolbox Home Page: <http://toolbox/Orleans>
* Horton Toolbox Home Page: <http://toolbox/Horton>

1. See the “Running Orleans on Windows Azure” document for more background on how this sample app is configured. [↑](#footnote-ref-1)