



# TwittZure

## Silverlight Twitter Client on Azure

November 20, 2009

Prepared for:  
Internal

Contact:  
Anthony Baker  
[abaker@schematic.com](mailto:abaker@schematic.com)

Yvette Pasqua  
[ypasqua@schematic.com](mailto:ypasqua@schematic.com)

### Los Angeles

3457 S. La Cienega Blvd, Los Angeles, CA 90016  
Phone 310-202-2900 / Fax 310-202-2910

### Atlanta

10 Tenth Street NE, Suite 250, Atlanta, GA 30309  
Phone 678-651-2800 / Fax 678-651-2802

### Minneapolis

10 South 5th Street, Suite 7017, Minneapolis, MN 55402  
Phone 612-305-4400 / Fax 612-305-4407

### London

40 Strand London, UK WC2N 5RF  
Phone 020 7969 4546 / Fax 020 7969 4000

### New York

41 East 11th Street, 6th Floor, New York, NY 10003  
Phone 212-710-2400 / Fax 212-710-2499

### Austin

206 East 9th Street, 16th Floor, Austin, TX 78701  
Phone 512-691-0481 / Fax 512-597-2546

### San Francisco

60 Green Street, San Francisco, CA 94111  
Phone 415-365-6846 / Fax 415-365-3190

### San José

Oficentro Trilogía, Edificio #2, Oficina 224  
Guachipelin de Escazú, San José, CR  
Phone 011 506 2289 0606 / Fax 011 506 2588 1038

# Table of Contents

Table of Contents	2
1.0 – Project Overview	3
2.0 – Technology	4
3.0 – Features	5
4.0 – Development Team	7
5.0 – Additional Considerations	8
7.0 – Glossary	9

# 1.0 – Project Overview

**TwittZure** is a Silverlight 3 Twitter Client application that runs on Microsoft's Windows Azure Cloud platform.

The main goal is to provide users with a Silverlight 3 Twitter client application that takes advantage of Silverlight 3 capabilities for Rich Internet Applications to create a simple yet engaging user interface to communicate with the two existing Twitter public APIs and serve as a proof of concept of the capabilities of the Microsoft Windows Azure platform.

**TwittZure** public URL: <http://twittzure.cloudapp.net/>

This project has three main technical goals:

- 1.) **Windows Azure Cloud.** Using the Windows Azure Platform to host for the application is due to the increasing market and client interest in cloud solutions and the momentum cloud application integration is currently experiencing. Cloud platforms are not only good solutions for the social-sharing capabilities of applications but also as an alternative to expensive hosting setups of the past. The Windows Azure Cloud provides a scalable and robust application, services and data hosting platform for .NET applications.
- 2.) **Silverlight 3 Application.** Developing this application will expand our expertise building bleeding edge Silverlight applications and allow us to implement some of the new features. Some of the new Silverlight 3 features implemented are asynchronous API calls, Out of Browser capabilities, application states, 3D Perspectives, and rich text display.
- 3.) **Twitter APIs.** The Silverlight-Azure hosted application integrates with Twitter APIs. Twitter has two public APIs, one for search capabilities and another for user information. The application communicates with both Twitter APIs using different formats providing a successful proof of concept for integration with social-enabled online properties.

## 2.0 – Technology

The following technologies were used for the **TwittZure** implementation:

- 1.) **Silverlight 3.** Current version of Microsoft Silverlight RIA plug-in.
- 2.) **Windows Azure.** Cloud hosting platform for the **TwittZure** application. **TwittZure** is a Windows Azure Hosted Service application.
- 3.) **C#3.0.** Coding language used for the application implementation.
- 4.) **.NET Framework 3.5.** The application was built on top of the Microsoft .NET Framework 3.5
- 5.) **Linq.** Used for XML and Atom parsing when retrieving data from Twitter.
- 6.) **ASP.NET 3.5.** Used to serve as a bridge between the **TwittZure** application and the Twitter APIs communication to work around Silverlight 3 cross domain and authentication limitations.
- 7.) **Expression Blend.** User to develop the **TwittZure** UI and create smooth transitions between the different application states.
- 8.) **Twitter APIs.** Data source services for the **TwittZure** application.

# 3.0 – Features

The following features are available in the current **TwittZure** version:

## **In Browser Experience:**

- 1.) **Twitter Search.** Ability to enter a search term and retrieve all public updates/posts containing the search term entered.
- 2.) **Twitter User Authentication.** Ability to enter username and password and authenticate against the Twitter API.
- 3.) **User Information Retrieval.** Ability to retrieve user's Twitter information, including image, screen name, location, URL, followers count, friends count, and favorites count.
- 4.) **User Timeline retrieval.** Ability to retrieve user's Twitter home timeline including user and friends latest's updates.
- 5.) **Rich Text Display.** Ability to display rich text including inline hyperlinks when provided by the Twitter API and render HTML text.
- 6.) **Paginated Results.** All updates/posts are retrieved in a paginated fashion to reduce round trips and loading times.
- 7.) **Status Update.** Ability to post new updates from **TwittZure** for authenticated users.
- 8.) **Install State Detection.** Ability to detect if the application has been installed for the out of browser experience. Install button will be showed if application is not installed.
- 9.) **Running out of Browser Detection.** Ability to detect if the application is running out of the browser.

### Out of Browser Experience:

- 1.) **Twitter Search.** Ability to enter a search term and retrieve all public updates/posts containing the search term entered.
- 2.) **Twitter User Authentication.** Ability to enter username and password and authenticate against the Twitter API.
- 3.) **User Information Retrieval.** Ability to retrieve user's Twitter information, including image, screen name, location, URL, followers count, friends count, and favorites count.
- 4.) **User Timeline retrieval.** Ability to retrieve user's Twitter home timeline including user and friends latest's updates.
- 5.) **Rich Text Display.** Ability to display rich text including inline hyperlinks when provided by the Twitter API and render HTML text.
- 6.) **Paginated Results.** All updates/posts are retrieved in a paginated fashion to reduce round trips and loading times.
- 7.) **Install State Detection.** Ability to detect if the application has been installed for the out of browser experience.
- 8.) **Running out of Browser Detection.** Ability to detect if the application is running out of the browser. If the application is running out of the browser, no "install" button will be shown, and instead, network status and "check for updates" information will be shown.
- 9.) **Network Detection.** Ability to detect network connectivity changes regarding network availability. Information of network state will be shown only in the out of browser experience.
- 10.) **Updates.** Ability to check for updates and retrieve latest application version if available. "Check for updates" button and Information will be shown if running out of the browser.

# 4.0 – Development Team

**Anthony Baker**

Software Architect, Microsoft Platforms Group  
Application Architect, Project Lead, Developer

**Jose Madriz**

Senior Motion Designer, Advanced Interactive Group  
Animator

**Alejandro Rodriguez**

Associate Motion Designer, Advanced Interactive Group  
Designer and Animator

**Gustavo Apuy**

Art Director, Creative Services  
Art Director and Designer

## 5.0 – Considerations

**TwittZure** is in its BETA release and serves as a proof of concept to demonstrate the capabilities of Windows Azure + Silverlight + Social Networks. Currently, the application uses several ASMX pages hosted on Windows Azure to serve as a proxy between the Silverlight application and the Twitter API, providing a good work around for possible changes in the Twitter cross domain policies and also to solve the network credentials limitations of Silverlight 3 for user authentication. However the application implements basic authentication and the credentials being sent from the application to the API are not secured for public mass use. Eventually the application will implement a robust middle tier to provide communication services taking advantage of new Windows Azure features and also implementing a more secure communication schema for user authentication.

## 7.0 – Glossary

- **Silverlight 3.** Microsoft Silverlight is a web application framework with a scope similar to Adobe Flash. The final version was released on July 9, 2009. It is compatible with multiple web browser products used on Microsoft Windows and Mac OS X operating systems. Mobile devices, starting with Windows Mobile 6 and Symbian (Series 60) phones, probably will be supported in 2010. A free software implementation named Moonlight, developed by Novell in cooperation with Microsoft, is available to bring compatible functionality to Linux. ([http://en.wikipedia.org/wiki/Microsoft\\_Silverlight](http://en.wikipedia.org/wiki/Microsoft_Silverlight))
- **Windows Azure Platform.** The Windows® Azure™ Platform (Azure) is an internet-scale cloud services platform hosted in Microsoft data centers, which provides an operating system and a set of developer services that can be used individually or together. Azure's flexible and interoperable platform can be used to build new applications to run from the cloud or enhance existing applications with cloud-based capabilities. Its open architecture gives developers the choice to build web applications, applications running on connected devices, PCs, servers, or hybrid solutions offering the best of online and on-premises.
- **Twitter APIs.** The Twitter API consists of two parts. This is entirely due to history. Summize, Inc. was originally an independent company that provided search capability for Twitter data. [Summize was later acquired and rebranded as Twitter Search](#). Rebranding the site was easy, fully integrating Twitter Search and its API into the Twitter codebase is more difficult. It is in our pipeline to unify the APIs, but until resources allow the REST API and Search API will remain as separate entities. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user information. The Search API methods give developers methods to interact with [Twitter Search](#) and trends data. The concern for developers given this separation is the effects on [rate limiting](#) and output format.