

Facebook IE Toolbar

Design Specification

Authors: Sergey Lavrinenko,

Edward Yablonsky,

Yan Drugalya

Lohika

2121 El Camino Real #100

San Mateo

CA 94403

**Table of Contents**

Contents

[1. Revision History 3](#_Toc221184098)

[2. Introduction 4](#_Toc221184099)

[2.1. Scope 4](#_Toc221184100)

[2.2. Intended Audience 4](#_Toc221184101)

[2.3. Related Documents 4](#_Toc221184102)

[3. Design Requirements 5](#_Toc221184103)

[3.1. Installation Requirements 5](#_Toc221184104)

[3.2. Toolbar Functional Requirements 5](#_Toc221184105)

[3.3. Non-Functional Requirements and Technical Constraints 5](#_Toc221184106)

[3.4. Documentation and Installation Requirements 5](#_Toc221184107)

[4. Design Model 7](#_Toc221184108)

[4.1. FBIEToolbar.dll Component Design 7](#_Toc221184109)

[4.1.1. Toolbar and Sidebar Design 8](#_Toc221184110)

[4.2. FBClientService.exe Component Design 9](#_Toc221184111)

[4.2.1. Information Update Logic Implementation 10](#_Toc221184112)

[5. Installation 11](#_Toc221184113)

[6. Risks and Contingencies 12](#_Toc221184114)

# Revision History

|  |  |  |
| --- | --- | --- |
| **DATE** | **REVISION** | **DESCRIPTION OF CHANGES** |
| 23 Jan 2009 | 0.1 | First Draft |
| 30 Jan 2009 | 1 |  |
|  |  |  |

# Introduction

## Scope

This document presents a detailed solution to the problem defined by the Facebook IE Toolbar Software Requirements Specification.

## Intended Audience

This document is intended for use internally by Lohika, Inc. Development and Sustaining Engineering departments. It contains confidential information proprietary to Lohika, Inc. and is not to be copied, circulated, or substantively discussed without prior approval of Lohika, Inc. management.

## Related Documents

1. Facebook IE Toolbar SRS.docx
2. <http://wiki.developers.facebook.com/index.php/API>

# Design Requirements

This section contains summary of requirements described in [1].

## Installation Requirements

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Description** |
| INST.1 | Installation from Web page | User has ability to install toolbar under Internet Explorer easily. There is web page on which user can click link and following instruction of the installer get toolbar in their browser. |
| INST.2 | Installation from Windows installation |  |

## Toolbar Functional Requirements

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Description** |
| TLB.1 | Login/Logout | User should be able to login/logout to Facebook from toolbar. Appropriate html dialog will appear for login, which will request username and password and after that session would be established. |
| TLB.1.1 | Save credentials | User should be able to select “Save Password” checkbox to save his login credentials. When saved, Toolbar will automatically login to Facebook. |
| TLB.2 | Quick Links | There is a number of quick links available on the Toolbar: Profile, Friends, Networks, Inbox, Photos, Notes, Groups, Events, Posted Items, Account, Privacy, Home, Go to Profile. They should redirect browser to corresponding Facebook page. |
| TLB.3 | Set Status |  |
| TLB.4 | Get notified by Icons |  |
| TLB.5 | Get notified by popup window |  |
| TLB.6 | Share Content |  |
| TLB.7 | Search Facebook from Anywhere |  |
| TLB.8 | Facebook friends sidebar |  |
| TLB.8.1 | Sorting |  |
| TLB.9 | Toolbar Settings |  |

## Non-Functional Requirements and Technical Constraints

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Description** |
| NFR.1 | Look and Feel | Toolbar should be similar in UI to Firefox toolbar. |
| NFR.2 | Communication Interface to Facebook | Toolbar should use the same interface (HTTP) as original Firefox toolbar to interact with the Facebook system. |
| NFR.3 | Security Requirements | All personal information of the user should be protected to not allow other persons read it. |
| NFR.4 | System configuration supported | MS Windows XP/Vista  MS IE 7.0 |
| NFR.5 | Development language/environment | C++, MS VS 2008 |
| NFR.6 | Code conventions | All code should follow Facebook C++ Coding Conventions and C++ Style. |

## Documentation and Installation Requirements

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Description** |
| DIR.1 | Installation from Web page | User has ability to install toolbar under Internet Explorer easily. There is web page on which user can click link and following instruction of the installer get toolbar in their browser. |
| DIR.2 | Installation from Windows installation |  |
| DIR.3 | Documentation | The final product should be documented both inline as well as externally in the same manner the current Firefox toolbar is. |

# Design Model

High Level Architecture is shown on the figure below.



Toolbar and Sidebar are implemented as COM components with interfaces defined by Microsoft for IE Toolbars. They implemented inside COM Inproc Sever FBIEToolbar.dll. Toolbar and Sidebar components obtain and update all Facebook information (user details and status, his friends’ details) using FBClientServices.exe. FBClientServices.exe is a COM Outproc Local server, which implements all logic for communication and information retrieval/uploading from/to Facebook server. Such architecture allows keeping all Toolbars synchronized and updated simultaneously, and minimizes network traffic between client and server.

## FBClientService.exe Component Design



As stated above, FBClientService.exe is a COM Local Outproc Server, which implements COM component with IFBClientService interface. It sends notifications about data updates to its clients using dataUpdated event in IFBClientServiceEvents interface.

The component implements all logic of communication with Facebook server, update on timers, profile and session storage.

### Downloading/Uploading information from/to Facebook server

CommUtils utilities are a set of functions (either belonging to same class or independent), which implement logic or communication and information retrieval/uploading from/to Facebook server.

These utilities implement following functions:

1. Login to Facebook server
2. Calculate session key and session secret using MD5 algorithm to use them for following operations
3. Downloads from Facebook server information about logged-in user and number of events (pokes, messages etc) and fills it to instances of FacebookData and FacebookUserData classes
4. Downloads list of user friends, their attributes and fills instances of FacebookUserData class with this information
5. Downloads friend’s images from Facebook server, stores them to IE cache or any other convenient place. Full path to such file is kept in “pic“ attribute of FacebookUserData structure. By this implementation, we minimize network traffic – all images will be downloaded only one time. File destination and naming are TBD.

### User information

All information about current user, his status, number of different events etc is kept in instances of FacebookData and FacebookUserData classes. All information about user’s friends is kept as a collection of FacebookUserData objects as shown on the figure below.

FacebookUserData has same attributes as fbIFacebookUser interface, which is used in original Facebook Toolbar for FireFox. One important difference is that “pic” attribute is full path to file, where user’s image is stored by CommUtils as described in previous section.

### Settings

ServiceSettings class is responsible for keeping and storing/loading all toolbar settings and user credentials. It is implemented as Singleton.

Toolbar settings and user credentials can be stored User’s Application Data MS Windows directory. User credentials should be protected by some stable encryption algorithm, or alternatively, can be separately stored in MS Windows protected storage.

### Information Update Logic Implementation

Facebook information update is made every 5 minutes by timer. Communication with Facebook server is made by CommUtils utilities in a separate thread. CommUtils use HTTP as a communication protocol. Detailed information about the protocol and request/response formats can be found in [2].

CommUtils fill new instances of FacebookData and FacebookUserData classes. Then these new objects are compared with old ones and based on difference, ClientService class object sends corresponding dataUpdated events, one per each type of different information.

## FBIEToolbar.dll Component Design

As stated above, FBIEToolbar.dll is a COM Inproc server, which implements Toolbar and Sidebar COM components. Both Toolbar and Sidebar components should implement standard MS IE interfaces (IDeskBand, IObjectWithSite, IInputObject, IPersistStream) to interact with MS IE.

When created, Toolbar and Sidebar obtain instance of FBClientServices component, obtain from it all information presented to user and subscribe to FBClientServices update events. By these events, they get notifications when any information has been updated and update their UI views accordingly. See several samples of interactions in section 4.3.

Because any COM interaction is time consuming, to minimize it we create single event sink point, implemented in ClientServiceEventsSink class. Single instance of this class subscribes to FBClientServices update events and then redistribute these events to Toolbar and Sidebar.

In the same time, Toolbar and Sidebar obtain all information they require from FBClientServices independently.



### Toolbar and Sidebar Design

**Toolbar and Sidebar**

Toolbar and Sidebar general structure is shown on the figure below. They use standard and customized MFC controls to display their content.

**Share Content** **Window**

Share Content Window and its functionality are implemented as standard MS IE popup dialog window, redirected to corresponding Facebook page.

**Settings Dialog**

Settings Dialog is implemented as a standard modal MFC dialog with corresponding controls. It uses ServiceSettings class from FBClentService.exe for information retrieval and update.

**Friends List**

The most complex UI control is FriendsView which displays list of friends with names, pictures, statuses and links as in original FireFox toolbar. It is implemented as a ListView with user drawn ListViewItem and reused both in Sidebar and Toolbar (for search drop down).

The information for each ListViewItem is taken from FacebookUserData objects, friends’ images are loaded from disk and drawn using GDI+ library.



## Interaction Diagrams

Following diagrams show most complicated types of interactions between FBClientService.exe and FBIEToolbar.dll components.

### Create Toolbar

The diagram below shows the process of Toolbar creation, when new instance of IE is starting.



### Update

The diagram below shows method calls sequence executed, when Facebook user data is updated by timer. First diagram shows interactions going inside FBClientService.exe component and second one show interactions in FBIEToolbar.dll and FBClientService.exe, but with most focus on Tollbar logic.





# Installation

Installation will be implemented as standard Windows MSI installation.

Installation will include following steps:

* Copy all files to destination folder selected by user
* Register all COM components which require registration

# Risks and Contingencies

Following risks are identified:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Description** | **Risk** | **Impact** | **Mitigation** |
| 1 | FriendsListView implementation | There may be technical difficulties with implementation of its functionality to have same look and feel as in original toolbar. | Low | Schedule |  |
| 2 | Facebook communication protocol | The protocol seems to be simple, but session code uses MD5 algorithm and we had some problems with it on first try. | Medium | Schedule | Actually we have sample of its implementation in JS (Firefox plugin) and seems there is a code in Internet. Thus, recommendation is to make prototype first and start it asap. |
| 3 | Web page installation | All similar toolbars usually installed by standard Windows installation package. We haven’t found samples/or toolbars which can be installed directly from Web page. There is high possibility that there are technical limitations for such installation. Also, we suppose that there may be difficulties with upgrade in such scenario. | High | Scope/  Schedule | Propose to investigate problem. There is a high possibility that we’ll have to drop this part of functionality. |