



i2b2 Web Client Software

Install and Configuration Guide

Document Version: 1.1
i2b2 Software Version: 1.6

Table of Contents

DOCUMENT MANAGEMENT	3
INTENDED AUDIENCE.....	4
1. INTRODUCTION	5
2. REQUIREMENTS.....	6
2.1. BROWSER REQUIREMENTS	6
2.2. SERVER REQUIREMENTS	6
2.2.1. cURL.....	6
3. PROXY INFORMATION.....	7
4. DIRECTORY STRUCTURE.....	8
4.1. FRAMEWORK.....	8
4.2. HIVE AND CORE CELLS.....	8
4.3. MODULES AND PLUG-INS	8
5. INSTALL	9
5.1. DOWNLOAD THE I2B2 WEB CLIENT FILES	9
5.2. COPY FILES TO WEB SERVER DIRECTORY	11
6. CONFIGURING THE WEB CLIENT	12
6.1. PROXY CONFIGURATION	12
6.1.1. <i>Configure Security in PHP Proxy</i>	<i>12</i>
6.1.2. <i>Configure the Web Client to Connect to the Proxy.....</i>	<i>13</i>
6.1.3. <i>Hive Connection Configuration.....</i>	<i>14</i>
6.2. MODULE LOADER CONFIGURATION	16
6.3. PLUG-IN CONFIGURATION.....	17
7. LICENSE	21

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.0	09/18/08	Nick Benik and Griffin Weber, MD, PhD	Initial document written for i2b2 software version 1.3
1.1	08/06/12	Janice Donahoe	Updated for i2b2 software version 1.6

INTENDED AUDIENCE

This document was created as an overview on how to quickly install the i2b2 Web Client. The author(s) assume that the reader is an experienced software developer or architect who has some experience programming JavaScript and some knowledge of advanced JavaScript topics such as JSON, AJAX, and using JavaScript toolkit (APIs).

1. INTRODUCTION

The i2b2 Web Client is an alternative web-based interface to the i2b2 Hive, which reproduces much of the functionality of the desktop client through a web browser. The primary advantage of the Web Client is that the software does not have to be installed on the user's computer. This aids greatly in enterprise-wide rollout of the i2b2 across an institution. However, because of the current limitations of web browsers, certain advanced functionality will continue to require a subset of "power users" to download the desktop client.



"i2b2 Workbench" is the name of the desktop client which is packaged as an Eclipse application.

2. REQUIREMENTS

The requirements for installing and using the i2b2 Web Client are defined in the following sections.

2.1. Browser Requirements

The following web browsers are supported at this time:

Browser	Version(s)
Internet Explorer	6, 7, 8, and 9
Firefox	2 and 3
Safari	3

 ***JavaScript must be enabled.***

2.2. Server Requirements

The i2b2 Web Client can be hosted from any web server that supports HTML, JavaScript, CSS, and GIF / JPG / PNG image files. It has been tested on Apache for Linux and Windows, as well as Windows IIS.

2.2.1. cURL

The i2b2 Web Client requires cURL (Client URL Request Library) be installed on the server. You need to compile your Apache install with cURL enabled or host PHP on Windows IIS with cURL installed from the PHP Lib.

For more information on cURL you can look at these websites:

<http://en.wikipedia.org/wiki/CURL>

<http://www.php.net/manual/en/install.unix.apache2.php>

<http://www.php.net/manual/en/book.curl.php>

3. PROXY INFORMATION

The i2b2 Web Client communicates with server-side web services using AJAX (Asynchronous JavaScript and XML).

All modern web browsers follow the a universal security standard which does not allow JavaScript code to directly communicate with any server that is not the exact same base URL that was used to load the web page. This behavior is critically important in preventing a hacker's code from offloading your personal information to a random IP overseas; it also prevents web-based applications from directly engaging an enterprise's SOA web services.

The solution to this problem is to use a web proxy that forwards the request to the correct service and then returns the response back to the browser. This extremely simple cell enables the browser based web client to participate fully with the i2b2 Hive.

Although the Web Client is platform independent, a server side proxy has to be written for the type of server that is hosting it. Included in the i2b2 Web Client software is a proxy written in PHP.

4. DIRECTORY STRUCTURE

4.1. Framework

Your deployment of the i2b2 Web Client will have a directory which contains all the *JavaScript* code related to the i2b2 Web Client Framework. The default location of this directory is **/js-i2b2**.

The main directory structure is as follows:

Directory	Description
/js-i2b2	Root directory for the Cell
/js-i2b2/hive	Directory containing the generalized web client framework
/js-i2b2/cells	Directory containing each Cell's implementation code

4.2. Hive and Core Cells

Within each cell there are several directories that are used to store the cell's files. Although every cell is different, the described standard should be followed. The example below (utilizing the Ontology Cell) shows how a cell's files will be organized within a directory.

Directory	Description
/js-i2b2/cells/ONT	Root directory for the Ontology Cell; contains view controller and Ajax code.
/js-i2b2/cells/ONT/assets	Non-code files needed by the Cell such as HTML templates or images

4.3. Modules and Plug-ins

Each code module is free to create its own directory structure however you must place all CSS files and icon images within a subdirectory called **assets** in order for the web client's framework to be able to use them correctly.

5. INSTALL

5.1. Download the i2b2 Web Client Files

The source code for the latest version of the i2b2 Web Client can be downloaded from the i2b2 website.

1. Go to the i2b2 website (<http://www.i2b2.org>)
2. At the top of the page there is a navigation bar, click on “**Software**” to go to the *software page*.



3. Once the software page loads, scroll down to the section called **Downloadables**.
4. Click on the **Source Code** box.

Downloadables



5. Information about the latest version and a list of zip files available for downloading will appear.

6. Select the appropriate file to download.

Downloadables

**Client Software**
i2b2 Workbench
Download the Windows or Macintosh executable

**Client & Server**
i2b2 VMWare
Download the VMWare image to run both the client and server






**Source Code**
i2b2 Source
Download the source code and build the client and server from scratch


Source Code

Version: 1.6.05
Release Date: July 11, 2012
License: [i2b2 Open Source License](#)

The i2b2 Source is a collection of both client and server source code and can be used to build the i2b2 Server and Workbench from scratch. The source code files for the i2b2 Core Hive Cells, i2b2 Workbench and Web Client are listed below. Demo data and the current documentation can also be downloaded.

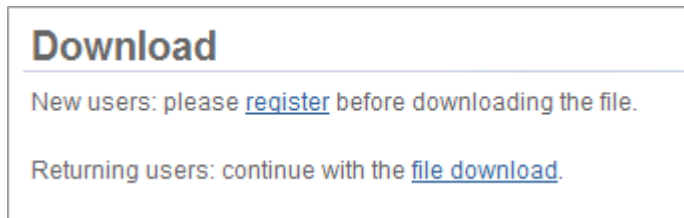
Select the appropriate file below for download:

Download	File Size	Type
i2b2core-src-1605.zip	45 MB	 Core Source Code Server v1.6.05
i2b2Workbench-src-1605.zip	72 MB	 Core Source Code Client v1.6.05
i2b2webclient-1605.zip	5.0 MB	 Web Client Source v1.6.05
i2b2createdb-1605.zip	20 MB	 i2b2 Data for Oracle and Sql Server
i2b2core-doc-1605.zip	12 MB	 i2b2 Documentation v1.6.05

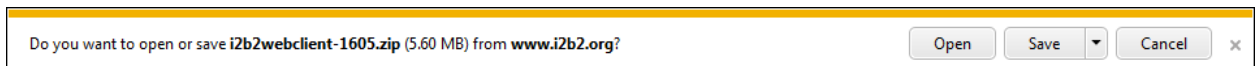
 **The name of the file shown in the above image may vary slightly depending on the version of i2b2 software you are downloading. The file name will always start with “i2b2webclient” and the last part of the name is dependent on the version of the software in the zip file.**

Example: “1605” means the zip file contains version 1.6.05 of the i2b2 Web Client.

7. The **Download** page will open. At this point, if you are not already a registered user you will need to do so. If you are a returning user you can click on the “**file download**” link.



8. The next step is to review the i2b2 license and choose to either accept or decline.
 - a. If you choose to **Decline** you will be brought back to the main i2b2 Software page.
 - b. If you agree with the license click on **Accept** to continue with the download.
9. The **Save dialog box** will open.



10. Click on the **Save** button and the zip file will be saved to your *Download folder*.

 ***Depending on your browser and it's version you may need to select a target directory to save the zip file.***

5.2. Copy files to Web Server Directory

After downloading the latest version of the i2b2 Web Client the next step is to copy the files to your web server directory.

1. Locate the zip file you just downloaded in the previous section.
2. Copy the zip file to your web server.
3. Unzip the file to your target directory.

6. CONFIGURING THE WEB CLIENT

Three levels of configuration exist within the i2b2 Web Client.

1. Configuration of web proxy and connection to various Hive PM Cells.
2. Registration and configuration of code modules (plug-ins / cells).
3. Configurations for individual code modules.

All configurations are written in JSON (JavaScript Object Notation) and are dynamically loaded by the i2b2 framework using AJAX calls. The files can be modified using any text editor.

! *When editing the configuration files it is very important to keep the overall structure of the files.*

6.1. Proxy Configuration

6.1.1. Configure Security in PHP Proxy

Included in the code distribution is **index.php** which can act as a simple proxy for use in a PHP-enabled Apache web server. This proxy file requires PHP to be compiled with XML support (default setting in most Linux distributions). Within the PHP file are 2 arrays that are used to restrict all requests from connected web clients to only the IP addresses that the cells are located at.

Array	Description
\$WHITELIST	List of acceptable URL prefixes.
\$BLACKLIST	List of invalid URL prefixes.

EXAMPLE:

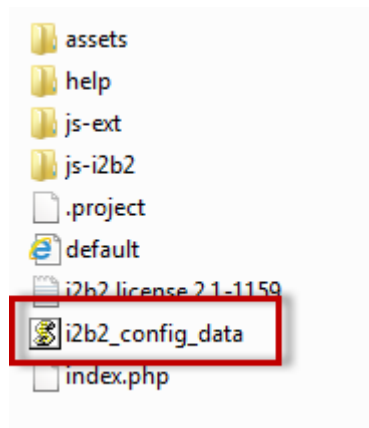
```
$WHITELIST = array(  
    "http:// ",  
    "http://127.0.0.1:9090/axis2/rest/ ",  
    "http://localhost:9090/axis2/rest/ ",  
    "http://127.0.0.1:7070/i2b2/rest/ ",  
    "http://localhost:7070/i2b2/rest/ ",  
    "http://services.i2b2.org ",  
    "https://services.i2b2.org "
```

```
);

$BLACKLIST = array(
    "http://127.0.0.1:9090/test/ ",
    "http://localhost:9090/test / ",
    "http://127.0.0.1:7070/test / ",
    "http://localhost:7070/test / "
);
```

6.1.2. Configure the Web Client to Connect to the Proxy

One of the most important configuration files is the *connection configuration* file called **i2b2_config_data.js**. It is located at the root directory of the distribution package.



The first configuration attribute **urlProxy** must be changed for every deployment. It should be set to point to *your* web proxy cell. In all cases it must be the same base URL (hostname and port) as the website that is used to load the initial **default.htm** file into the browser.

EXAMPLE:


```
{
    urlProxy: "index.php",
    urlFramework: "js-i2b2/",
    //-----
}
```

6.1.3. Hive Connection Configuration

In addition to the proxy information, the location of the files for the i2b2 framework and connections to one or more hives is also defined in the *connection configuration file* called **i2b2_config_data.js**.

Name	Null	Type	Description
urlProxy	N	String	The full path URL for the i2b2 web services proxy server.
urlFramework	N	String	The full path URL to the root js-i2b2 directory.
lstDomains	N	Array	An array containing 1 or more domain definition data objects.

The domain definition data object contains the following attributes for each data object:

Name	Null	Type	Description
domain	N	String	A short code used by the proxy server for the domain/group ID
name	N	String	A human-readable string containing the domain's name
urlCellPM	N	String	The full path URL for the Project Management Cell
allowAnalysis	Y	Boolean	<p>Defines whether or not the plugin viewer is available to users. The plugin viewer is used to access to the optional plug-ins (Analysis Tools).</p> <p>Y Analysis Tools will appear as an option on the menu bar. Users can access the optional plugins via the <i>plug-in viewer</i>.</p> <p>N Analysis Tools will not appear as an option on the menu bar. Users will not be able to access the optional plug-ins.</p> <p> <i>This setting will all affect all users.</i></p>
debug	Y	Boolean	Are debugging messages logged? (uses additional memory).

EXAMPLE:

The following example file shows the setup for multiple domains.

```
{
  urlProxy: "http://localhost/services/proxy/Service.asmx/webProxy",
  urlFramework: "js-i2b2/",
  //-----
  // THESE ARE ALL THE DOMAINS A USER CAN LOGIN TO
  IstDomains: [
    { domain: "HarvardDemo",
      name: "i2b2.org (1.5)",
      urlCellPM: "http://webservices.i2b2.org/i2b2/rest/PMService/",
      allowAnalysis: true,
      debug: false
    },
    { domain: "demo",
      name: " i2b2.org (1.6)",
      urlCellPM: "http://192.168.0.120:9090/i2b2/rest/PMService/",
      allowAnalysis: true,
      debug: false
    },
  ],
}
//-----
}
```

Using the above example, the user will see the following connection domains (i2b2 Host) when logging into the i2b2 Web Client.



The screenshot shows the 'i2b2 Login' window. It contains three input fields: 'Username:' with the value 'demo', 'Password:' with masked characters, and 'i2b2 Host:' which is a dropdown menu. The dropdown menu is open, showing three options: 'i2b2.org (1.6)', 'i2b2.org (1.6)', and 'i2b2.org (1.5)'. The background of the login window features a collage of medical images, including a brain scan, a heart scan, and a person in a hospital bed.

6.2. Module Loader Configuration

The framework is aware of various code modules after they are registered in the main component list configuration file. This list is located within the `/js-i2b2/i2b2_loader.js` file in a section containing *JSON-based configuration information* which has the following structure:

```
// THESE ARE ALL THE CELLS THAT ARE INSTALLED ONTO THE SERVER
i2b2.hive.tempCellsList = [
  { code: "PM",
    forceLoading: true // <--- this must be set to true for the PM cell!
  },
  { code: "ONT" },
  { code: "CRC" },
  { code: "WORK" },
  { code: "PLUGINMGR",
    forceLoading: true,
    forceConfigMsg: { params: [] }
  },
  { code: "Dem1Set",
    forceLoading: true,
    forceConfigMsg: { params: [] },
    roles: [ "DATA_LDS", "DATA_DEID", "DATA_PROT" ],
    forceDir: "cells/plugins/standard"
  },
  { code: "Dem2Set",
    forceLoading: true,
    forceConfigMsg: { params: [] },
    roles: [ "DATA_LDS", "DATA_DEID", "DATA_PROT" ],
    forceDir: "cells/plugins/standard"
  },
  { code: "Timeline",
    forceLoading: true,
    forceConfigMsg: { params: [] },
    roles: [ "DATA_LDS", "DATA_DEID", "DATA_PROT" ],
    forceDir: "cells/plugins/standard"
  },
  { code: "ProjectRequest",
    forceLoading: true,
    forceConfigMsg: { params: [] },
    roles: [ "DATA_LDS", "DATA_DEID", "DATA_PROT" ],
    forceDir: "cells/plugins/standard"
  }
];
```

This JSON structure is used to register a list of cells / plug-ins that are able to be loaded if the user has been authorized to use them (via the data returned from the *Project*

Management cell during successful login). The above code listing has information for registering the following cells / modules (in order):

1. Project Management Cell (forced to automatically load when the framework is loaded)
2. Ontology Cell
3. Data Repository Cell
4. Plugin Viewer Module (used to manage all non-cell plug-in modules)
5. Demographics (1 Patient Set) – Simple Counts
6. Demographics (2 Patient Sets) – Simple Counts
7. Timeline
8. Project Request

Unless a custom module is going to be running as an i2b2-compliant Cell module, further configuration options must be included in this file using the “forcing” options below:

Configuration Option	Description
forceLoading: (Boolean)	Is the module automatically loaded during framework initialization
forceConfigMsg: (Object)	This data object is automatically populated to i2b2.CELLCODE.cfg.config when the module is loaded

These options can also be used to override configuration information that is being returned to the web client Framework from the Project Management Cell during login authorization. The purpose of the **forceConfigMsg** setting is that it will force information to be automatically loaded and will create a configuration value for later use.

6.3. Plug-in Configuration

For the framework to be able to properly load a plug-in module, information which defines the new module must be provided. This is accomplished by creating a JSON-based configuration file within the plug-in’s root directory. An example would be located at:

/js-i2b2/cells/plugins/examples/ExampHello/cell_config_data.js

This file would have the following structure:

```
// this file contains a list of all files that need to be loaded dynamically for
```

```

this plugin
// every file in this list will be loaded after the plugin's Init function is
called
{
  files:[ "ExampHello.js" ],
  css:[ "ExampHello.css" ],
  config: {
    // additional configuration variables that are set by the system
    short_name: "Hello World",
    name: "Example #1 - Hello World",
    description: "This plugin demonstrates how to register a plugin within
the i2b2 web client framework.",
    category: ["celless","plugin","examples"],
    plugin: {
      isolateHtml: false,    // this means do not use an IFRAME
      isolateComm: true,    // this means to expect the plugin to use AJAX
communications provided by the framework
      html: {
        source: 'injected_screens.html',
        mainDivId: 'ExampHello-mainDiv'
      }
    }
  }
}
}

```

The configuration file has three main parts to it:

1. A list of JavaScript files
2. A list of HTML CSS files
3. A configuration section

The files and CSS configuration sections are self-explanatory. All filenames listed will be prepended with the plug-in's base directory to generate the full file access location used by the framework in loading the files.

! It is important to note that all version of Microsoft Internet Explorer limit the number of dynamically loaded style sheets to a total of 31 files. The web client framework uses several style sheets and each cell or plug-in may have dynamically loaded style sheets as well.

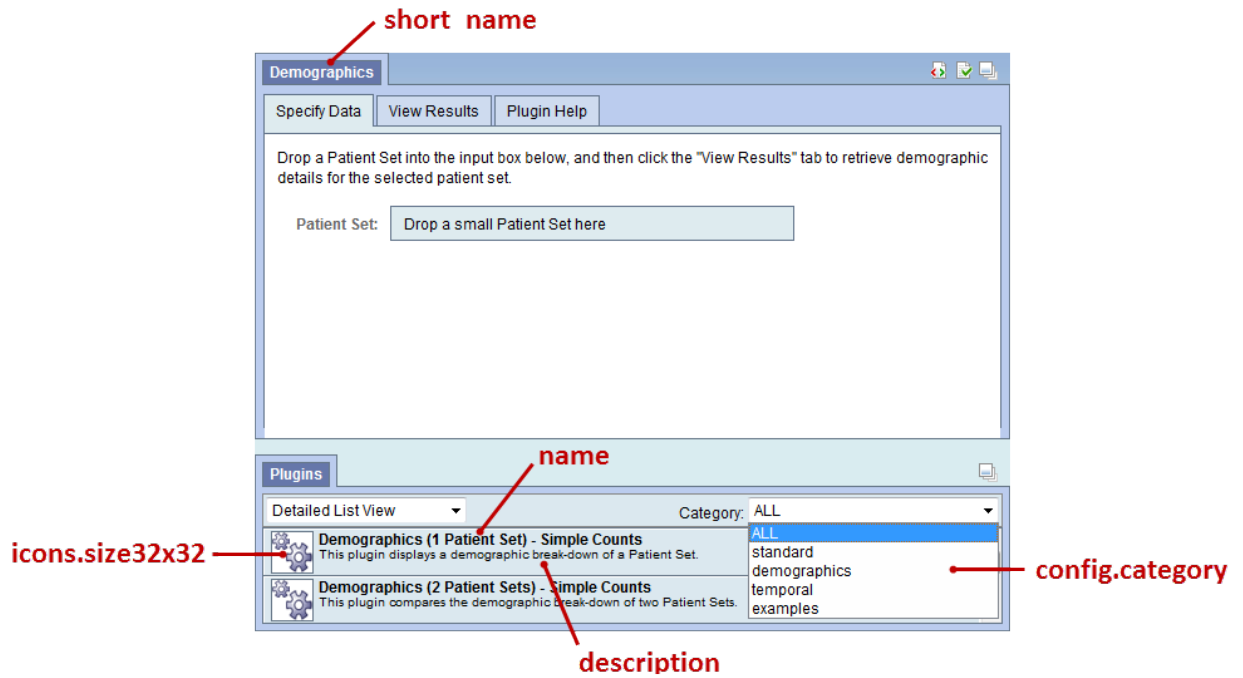
An important best practice is to only define one CSS file in your plug-in's configuration file.

To use more than one CSS file in your plug-in, create a CSS file to subsequently load your other CSS files using the @import (file.css) command.

The configuration section contains various pieces of information that are used by the framework. They are explained below:

JSON Configuration Variable	Description
config.short_name	Is displayed in the title tab area of the plug-in viewer's display window.
config.name	The title string that is displayed in the plug-in viewer's listing window.
config.description	The description that is displayed in the plug-in viewer's listing window.
config.category	A list of categories that this plug-in is a member of. All plug-ins must include "plugin" value. If the plug-in does not have a backend cell of its own then "celless" value should also be present.
config.icons	JSON object defining one or more icon files. These files must be located in the assets directory of the plug-in's base directory.
config.icons.size32x32	Filename for 32x32 pixel icon used in the plug-in listing window when in detailed view mode. (The fill must be in the plug-in's assets directory).
config.icons.size16x16	Filename for 16x16 pixel icon used in the plug-in listing window when in summary view mode. (The fill must be in the plug-in's assets directory).
config.plugin	JSON object which defines and configures the module as a plug-in.
config.plugin.isolateHtml	Boolean, should the framework isolated the plug-in's HTML an IFRAME.
config.plugin.html	JSON object that contains information about the plug-in's display HTML.
config.plugin.htm.source	Filename for the plug-in's display HTML (in the local assets directory).
config.plugin.html.mainDivId	The unique ID of the HTML Element (in the above declared source file) whose contents will be initially displayed.

Most of the information within the configuration section is used by the plugin viewer subsystem in ways that are reflected in the user interface.



7. LICENSE

The i2b2 source code is licensed under the i2b2 Software License 2.1. This includes but is not limited to all code in the edu.harvard.i2b2.* package namespace.

The i2b2 Web Client uses three open source JavaScript Libraries: Yahoo! User Interface (YUI), Prototype, and Firebug.

The js-ext directory includes the code for these, along with their respective license agreements.