



# ASP.NET Ajax Chat

## Table of Contents

<b>Deployment &amp; Integration</b>	<b>1</b>
Server Requirements	1
Standalone Installation	1
DotNetNuke 4 Integration	2
<b>Integration with an existing site</b>	<b>3</b>
Basic Integration	3
Integration with an existing user database	3
Integration with 3rd party avatars	5
Integration with 3rd party profiles	5
Instant Messenger Integration	6
<b>Audio &amp; Video streaming</b>	<b>8</b>
Streaming with FMS	8
Streaming with RED5	8
<b>Index</b>	<b>a</b>

# 1 Deployment & Integration

## Deployment & Integration

**AspNetAjaxChat** is very easy to install and use. You can also integrate it with your own ASP.NET applications. Pre-made integration for popular software such as DotNetNuke and AspNetDating is also available.

## 1.1 Server Requirements

### Server Requirements

1. Supported operating system
  - Windows XP Professional
  - Windows Server 2000
  - Windows Server 2003
  - Windows Server 2003 R2
2. Microsoft .NET Framework 2.0
3. Supported SQL Server
  - Microsoft SQL Server 2000
  - Microsoft SQL Server 2005
  - Microsoft SQL Express
4. Supported RTMP server (required only for audio/video streaming)
  - Macromedia Flash Media Server
  - Red5 Open Source Flash Server
  - Wowza Media Server (coming soon)

## 1.2 Standalone Installation

### Standalone Installation

The following guide shows the steps to install AspNetAjaxChat as standalone application. When installed in standalone mode the software uses internal room management and allows your visitors to instantly pick a username and chat.

You can download the standalone package from here:

[http://www.aspnetajaxchat.com/AspNetAjaxChat\\_Trial.zip](http://www.aspnetajaxchat.com/AspNetAjaxChat_Trial.zip)

1. Extract the archive to a virtual directory or web site. The virtual directory should be configured for ASP.NET 2.0 from IIS.
2. Create and configure the database for AspNetAjaxChat

- Create a database (e.g. "AjaxChat"). You can also use existing database.
- Configure the database connection string in the **web.config** file.

```
<connectionStrings>
  <add name="AjaxChat" connectionString="Data
Source=(local);database=AjaxChat;Integrated Security=True"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

- Open SQL Server Management Studio (or Query Analyzer for SQL Server 2000) and execute the SQL scripts located within the **SQL** folder against the database you've created.

---

## 1.3 DotNetNuke 4 Integration

### Installing Asp.Net Ajax Chat for DotNetNuke 4

The following guide shows you how to integrate AspNetAjaxChat into your DNN 4 installation.

You can download the DNN integration package from here:

[http://www.aspnetajaxchat.com/AspNetAjaxChat\\_DNN.zip](http://www.aspnetajaxchat.com/AspNetAjaxChat_DNN.zip)

1. Open your DNN installation and log in as **host**.
2. Open **Host > Module Definitions** and click on **Install New Module**.
3. Select the **AspNetAjaxChatDNN.zip** file and click on **Install New Module**.
4. You should see a page with detailed information about the installed files. If everything is ok you should see "**EndJob Installation successful.**" at the bottom of the page.
5. The **AspNetAjaxChat** option will be available on the module options and you can add it to any page.

## 2 Integration with an existing site

### Integration with an existing site

The following topics will show you how to integrate AspNetAjaxChat with your site

## 2.1 Basic Integration

### Basic Integration

This topic covers the most basic AspNetAjaxchat integration.

1. Extract AspNetAjaxChat in a sub folder of where your application is installed. If your site is located in c:\inetpub\wwwroot\YourSite you can extract the chat in c:\inetpub\wwwroot\YourSite\AjaxChat.
2. Move the contents of the **AjaxChat\bin** folder to your site **bin** folder.
3. Move the contents of the **AjaxChat\App\_themes** folder to your site **App\_themes** folder.
4. Create the database as explained in Standalone Installation (see page 1)
5. Open the web.config file and add a connection string for AspNetAjaxChat (in the example below replace the sample one with your actual connection string)

```
<connectionStrings>
  <add name="AjaxChat" connectionString="Data
Source=(local);database=AjaxChat;Integrated Security=True"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

As an alternative to step 5 you can implement the IHttpApplicationConnectionStringProvider interface in your Global.asax (.cs/.vb) file.

This covers the basic integration.

## 2.2 Integration with an existing user database

### Integration with an existing user membership database

In this topic we will explain the integration with an existing user membership database. Once the integration is completed your users will be able to chat using the usernames they already have when they registered in your site. In order to proceed with this integration you should have completed the Basic Integration (see page 3).

1. Add the **AjaxChat.dll** to your project references.
2. Open **Global.asax(.cs/.vb)** and implement the **IHttpApplicationUserAdapter** interface. In order to implement the interface you need to implement the following methods:

```
string GetCurrentlyLoggedInUsername();
bool IsRoomAdmin(string username, int chatRoomId);
```

```

bool HasChatAccess(string username, int chatRoomId);
string GetUserDisplayName(string username);
bool UserExists(string username);
string GetLoginUrl();

```

Here's an explanation for each of the methods:

**string GetCurrentlyLoggedInUsername()** - This method is called when the chat needs to know which user is currently logged in. When implementing you should return the username of the currently logged in user. Return *NULL* if there is no logged in user.

```

public string GetCurrentlyLoggedInUsername()
{
    if (HttpContext.Current != null && HttpContext.Current.Session["UserSession"] != null)
    {
        UserSession sess = (UserSession)HttpContext.Current.Session["UserSession"];
        return sess.Username;
    }
    else
    {
        return null;
    }
}

```

**bool IsRoomAdmin(string username, int chatRoomId)** - This method is called when the chat needs to know if the specified user is administrator for the specified room. Administrators can kick and ban users and can also change the topic. Return *true* or *false* depending if the user is administrator.

```

public bool IsRoomAdmin(string username, int chatRoomId)
{
    return username == "admin";
}

```

**bool HasChatAccess(string username, int chatRoomId)** - This method is called when the chat needs to determine whether the user has access to the chat room. Use this method when you need to create private rooms available only for certain users or if the chat is only available for paid users, etc. If all users should have access to the chat simply return *true*.

```

public bool HasChatAccess(string username, int chatRoomId)
{
    // All members have access to the main chat room
    if (chatRoomId == 0)
        return true;

    // Check if the member has access to the chat room (custom logic here)
    GroupMember groupMember = GroupMember.Fetch(chatRoomId, username);
    if (groupMember == null || !groupMember.Active)
        return false;
    else
        return true;
}

```

**string GetUserDisplayName(string username)** - Use this method if you want the chat display name to be different from the username. Otherwise simply return the username.

```

public string GetUserDisplayName(string username)
{
    return username;
}

```

**bool UserExists(string username)** - The software uses this method to determine if the specified user exists in the user database. Return *true* or *false* depending on whether the user exists or not.

```

public bool UserExists(string username)
{
    return MyClass.CheckIfUserExists(username);
}

```

**string GetLoginUrl()** - This method should return the url of the page where your users can log in. AspNetAjaxChat will redirect non-logged in users to the specified url.

```
public string GetLoginUrl()
{
    return "http://www.yoursite.com/login.aspx";
}
```

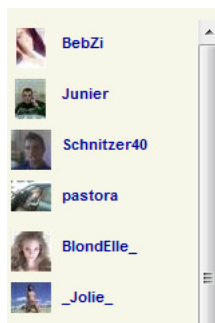
That covers the integration with an existing user database. For more advanced integration check the next chapters.

---

## 2.3 Integration with 3rd party avatars

### Integration with 3rd party avatars

If your site has forum or you are running dating/community site then your users most likely have a photo or avatar for their account. With AspNetAjaxChat your users can use their avatar in the chat! Here's an example:



In order to use your avatars with AspNetAjaxChat you need to implement the **IHttpApplicationSupportAvatars** interface. There is just one method to implement - **string GetUserAvatar(string username)**. The method should return the url of the image that will be used as avatar. The default themes provided with AspNetAjaxChat are designed for avatars that are sized 30x30. If you plan using bigger avatars you might need to adjust the design accordingly.

```
public string GetUserAvatar(string username)
{
    return MyClass.GetUserThumbnailUrl(username);
}
```

This covers the integration with 3rd party avatars. You can also use the functionality to specify different icons for males, females, paid, unpaid users, etc.

---

## 2.4 Integration with 3rd party profiles

### Integration with 3rd party profiles

If you are running dating/community site then your users most likely have a profile page with information and photos about them. With AspNetAjaxChat your users can see those profiles simply by clicking on the user avatar!

In order to use your profiles with AspNetAjaxChat you need to implement the **IHttpApplicationSupportProfiles** interface. There is just one method to implement - **string GetUserProfileUrl(string username)**. The method should return the url of the profile of the specified user.

```
public string GetUserProfileUrl(string username)
{
    return "http://www.mysite.com/profile.aspx?uid=" + username;
}
```

```
}
```

This covers the integration with 3rd party profiles.

---

## 2.5 Instant Messenger Integration

### Instant Messenger Integration

This topic covers the integration of the ASP.NET Instant Messenger. In order to proceed with this integration you should have completed the Basic Integration (☞ see page 3) and the Integration with an existing user database (☞ see page 3).

The Instant Messenger (IM) consists of two parts:

1. The **Presence** component - it maintains a list of currently online users and shows the "User XXX wants to chat with you..." dialog when chat is initiated.
2. The **MessengerWindow** component - it is the actual messenger window that is used to chat.

### Installing the Presence component

The **Presence** component should be placed on every page that can be accessed by the logged in users. A common place is the header or the footer of the site. The component file is **Components\MessengerPresence.ascx**. You can add it to your pages just like any other ascx components. For example:

- Add the following component registration directive in the header of the ASPX page (or master page):

```
<%@ Register Src="AjaxChat/Components/MessengerPresence.ascx" TagName="MessengerPresence"
    TagPrefix="uc2" %>
```

- Add the following code somewhere in the page:

```
<uc2:MessengerPresence ID="MessengerPresence" runat="server" />
```

At this point the **Presence** component should work and it should maintain a list of online users. It will also be loaded on all pages and if someone tries to chat the current user the **Presence** component will display an accept/reject dialog.

**Hint:** If your application implements the **IHttpApplicationSupportProfiles** interface then the dialog will also show link to the user profile. If your application implements the **IHttpApplicationSupportPhotos** interface then the dialog will also show a photo of the user requesting the chat.

### Setting up the "Start Chat" links

After the **Presence** component is installed all you need to do is place the **"Start Chat"** links. Usually you would want to place that link on the user profile - for example if you are viewing the profile of **John** you should be able to click **"Start Chat"** and that will launch a private chat with **John**. Assuming that the target username will be **"John"** the link will look like that:

```
<a href="AjaxChat/MessengerWindow.aspx?init=1&target=John" target="_blank">Start Chat</a>
```

Of course, replace **"John"** with the actual username.

If you want to specify the size of the window that opens you can use some javascript. For example:

```
<a href="#" onclick="window.open('AjaxChat/MessengerWindow.aspx?init=1&target=John',  
'ajaxim_John', 'width=650,height=400,resizable=1,menubar=0,status=0,toolbar=0'); return  
false;">Start Chat</a>
```

**Hint:** If your application implements the **IHttpApplicationSupportPhotos** interface then the chat window will show photos of both users. If you have configured the audio & video streaming then users will be able to video chat.

## 3 Audio & Video streaming

### Audio & Video Streaming

The following topics will show you how to enable the audio & video streaming for the various supported RTMP servers.

## 3.1 Streaming with FMS

### Streaming with Flash Media Server

This topic covers the a/v streaming when using Flash Media Server (FMS).

1. Download and install FMS. At the time of writing you can get the server from [here](#). There is also a free developer edition to get you started.
2. After FMS is installed create a folder called "ajaxchat" under the FMS application folder (e.g. *C:\Program Files\Macromedia\Flash Media Server 2\applications\ajaxchat*)
3. Open the AjaxChat admin tool (e.g. *http://localhost/AjaxChat/admin/home.aspx*). Log in (by default the username and password are "admin"). Open the "Settings" screen and enable the "**Enable Video Chat**" option.
4. Specify the url to your FMS in the "**Flash Media Server**" field (e.g. *rtmp://localhost/ajaxchat*).

That's it! Now you should be able to send and receive audio/video chats.

## 3.2 Streaming with RED5

### Streaming with RED5

This topic covers the a/v streaming when using the open source RED5 server.

1. Download and install RED5 (<http://osflash.org/red5>).
2. After RED5 is installed create a folder called "ajaxchat" under the RED webapps folder (e.g. *C:\Program Files\Red5\webapps\ajaxchat*)
3. Open the AjaxChat admin tool (e.g. *http://localhost/AjaxChat/admin/home.aspx*). Log in (by default the username and password are "admin"). Open the "Settings" screen and enable the "**Enable Video Chat**" option.
4. Specify the url to your RED5 in the "**Flash Media Server**" field (e.g. *rtmp://localhost/ajaxchat*).
5. Open the **web.config** file and set the server type to **red5**. The setting should look like this:

```
<appSettings>
  <add key="FlashServerType" value="red5"/> <!-- Valid values are "fms" and "red5" -->
</appSettings>
```

That's it! Now you should be able to send and receive audio/video chats.

## Index

### A

Audio & Video streaming 8

### B

Basic Integration 3

### D

Deployment & Integration 1

DotNetNuke 4 Integration 2

### I

Instant Messenger Integration 6

Integration with 3rd party avatars 5

Integration with 3rd party profiles 5

Integration with an existing site 3

Integration with an existing user database 3

### S

Server Requirements 1

Standalone Installation 1

Streaming with FMS 8

Streaming with RED5 8