FlexAuth “Getting Started” User Guide

(The following graphics are screen shots from Microsoft® ISA Server 2004 which is the property of Microsoft Corp. and are included here for instructive use. Some images illustrate FlexAuth, which is the property of Collective Software.)

Table of Contents

Problem Statement

Your organization has one or more web sites published through ISA 2004 Web Listeners and web publishing rules. The following limitations will apply:

- Each Listener will cause a separate authentication prompt, even if the same credentials might be valid across several Listeners. There is no way to implement single-sign-on functionality natively in ISA 2004.
- Your authentication options are limited to integrated Active Directory for user/group level controls, or RADIUS if your ISA servers are not domain members. With RADIUS you do not have the ability to assign permissions based on AD groups.
- You may be able to use the built-in OWA Forms-Based-Authentication option for non-OWA servers, but customizing the form is not supported.
- You cannot accept both FBA and Basic authentication on the same Listener. This means you have to split your server publishing into separate Listeners. For example, you need one external server name for OWA users that will utilize FBA, and another one for OMA/ActiveSync users that require Basic authentication.

Solution Overview

Collective Software FlexAuth was designed to easily overcome all these limitations with one convenient authentication filter. FlexAuth provides the following features:

- Seamless single-sign-on across several web listeners, provided they share an Internet domain suffix (such as *.example.com).
- Fully customizable Forms-Based-Authentication. Integrate the look and feel of your organization without breaking a sweat.
- Choose what client browser types will receive FBA screens; all other clients will automatically default to Basic Authentication.
- LDAP is supported as an authentication method, allowing your ISA servers to authenticate against an AD domain even if they are not domain members. LDAP TLS/SSL is also supported for secure communications.
- By using either Integrated AD authentication or LDAP, you can apply permissions to your publishing rules based on Windows groups and users.
- FlexAuth supports both ISA 2004 Standard and ISA 2004 Enterprise editions.

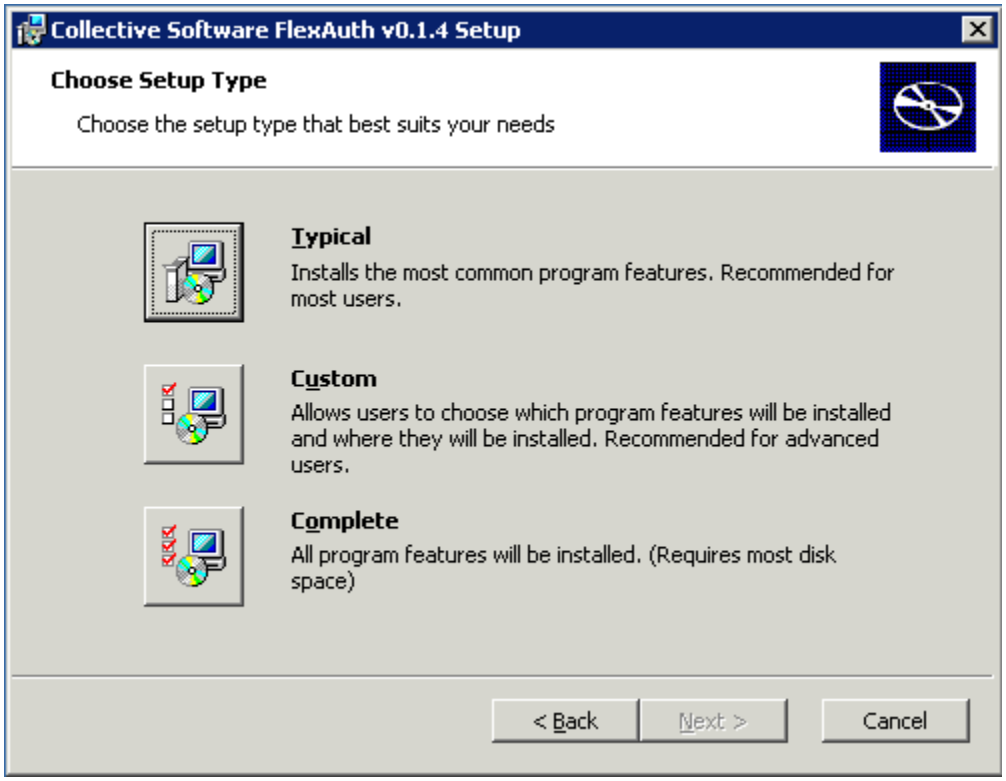
In the following sections, we will walk through the installation and configuration of FlexAuth for a common scenario that exercises all of the above features:

publishing OWA with FBA and supporting ActiveSync on the same listener, applying credentials in a single-sign-on fashion across several listeners, and using LDAP as an authentication method.

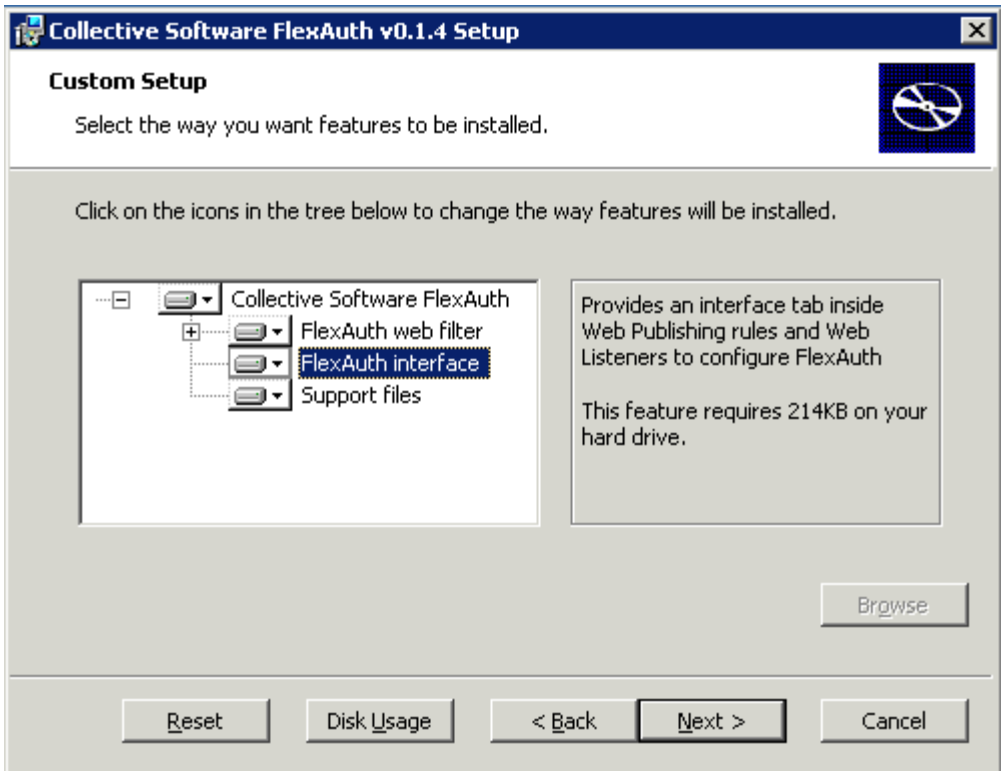
Help is available!

We are always happy to help you get our software set up and working. If you have questions or need assistance understanding/configuring/testing a Collective product, you can get in touch with our support staff quickly and easily. For the most up-to-date information, please see our Support page at http://www.collectivesoftware.com/Support/

Installation



The FlexAuth installer's "Typical" settings assume that your ISA installation is in the default location (C:\Program Files\Microsoft ISA Server) and that you wish to install both the Web Filter and the User Interface components on the server. **You can change any of these items by selecting the "Custom" install mode:**



You *must* install at least the “FlexAuth web filter” component on the ISA server itself. If you are using Enterprise Edition, this installer must be run separately on each of the ISA servers in your array. **You must complete the installer on the first array member before the rest of the members will be able to install.** If you attempt to install out of order, you'll get a message stating the above, and the install will roll back.

You should install the “FlexAuth interface” on all machines from which you will administer your ISA enterprise. This component extends the ISA console’s “Web Listener” properties dialog and allows you to configure the FlexAuth filter’s functionality.

In the event of install difficulties, the Windows Event Log (Application section) will usually contain more information about the problem, and should be sufficient to resolve the issue in most cases.

NOTE: In most cases the installation will complete without requiring a restart. If a restart is needed, the Windows Installer should automatically let you know. In some cases,

Configuring Single-Sign-On Realms

Accessing FlexAuth properties

FlexAuth adds a tab to the Web Listener properties dialog, which can be accessed from the Toolbox as shown here.

Please note that due to a limitation of the ISA console, accessing the Listener properties from the Publishing Rule properties dialog will *not* show the FlexAuth tab. The FlexAuth tab is only accessible in the manner shown above.

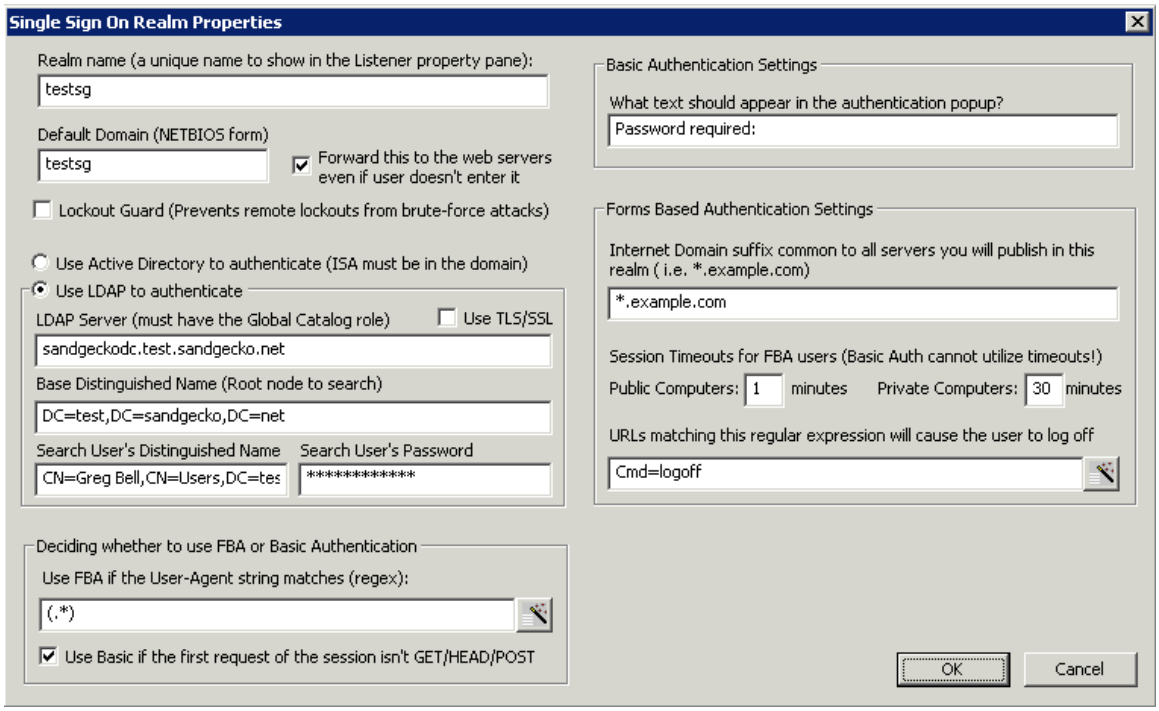
Creating your first realm

Choose a Listener that you wish to extend with FlexAuth, and go into the FlexAuth tab of its properties. Select the checkbox to enable FlexAuth on the Listener, and then the Single Sign On Realm selections will become active.

In order for the Listener to participate in a single sign on realm, you must select one from the dropdown. Of course when you start, the dropdown is empty because you haven't defined any realms yet.

Select “New” to begin configuration of a new realm.

The first thing to do is choose a name to refer to this realm. All Listeners that will participate in this single-sign-on realm will need to have the realm name selected in their dropdown boxes (shown above).



Default Domain

Enter the NETBIOS name name of your domain. This is used as a default in case the user does not enter a domain along with their username. By default, FlexAuth will forward this default domain to your web servers (in the form domain\username) if the user does not enter any domain. You can suppress this behavior by unchecking the relevant checkbox. If the user enters a domain or enters their credentials as a UPN format name, then their exact entry will be forwarded to the upstream web server. If you configure your publishing rules not to forward basic credentials (see Forward Basic credentials) then no authentication information will be passed to your target web servers.

Lockout Guard

Ordinarily, when you publish web servers to the Internet, you create a potential denial-of-service scenario. If a malicious individual knows one or more user names of your personnel, they can attempt to authenticate several times with bogus passwords, thereby locking out the user's account from the real owner, and causing administrative difficulties for you. FlexAuth can protect you from this type of attack with “Lockout Guard”. This technology will stop *one authentication attempt before lockout would occur*, so that it is, in general, not possible to lock out an account via FlexAuth. It should be noted that this system is not 100% fool proof due to the way in which Active Directory domain controllers store the “bad password count” data. However in ordinary scenarios it should provide effective relief against risk of lockout. To use Lockout Guard, check the appropriate box on the realm properties dialog.

When lockout guard is active, a user over the guard limit will continue to see the “login failed” message as though the uname/password was incorrect.

Integrated Auth

If you are using built-in Active Directory authentication (i.e. your ISA server is a domain member) then select the proper radio button.

LDAP Auth

If your server is not in the domain, you will need to use LDAP for authentication instead. Enter the full host name or IP address of your LDAP server (it should be a domain controller with the Global Catalog role). The Base DN should be the distinguished name string of where you want to search for users in your directory. This is usually just the root of your domain. So if your domain name is:

`"domain.example.com"`

then the correct Base DN would be:

`"DC=domain,DC=example,DC=com".`

The Search DN must be filled out with the distinguished name of a directory user who has permission to search the directory. So if your user's display name is:

`"Search User"`

and the user object resides in the default "Users" container, then the correct Search DN would be:

`"CN=Search User,CN=Users,DC=domain,DC=example,DC=com".`

If the user is in an OU called "Service Accounts" instead, then the DN would be:

`"OU=Service Accounts,CN=Users,DC=domain,DC=example,DC=com".`

A full discussion of LDAP syntax conventions is outside the scope of this document.

Finally, you must enter the password of the search user, so that FlexAuth is able to bind to LDAP in order to perform its search operations. **Remember if you change this password in the domain you must update your FlexAuth configuration or else authentications will start to fail.**

LDAP Auth with TLS/SSL

If you wish to support LDAP connections over TLS/SSL, then you can check that box. Keep in mind that in order for this type of connection to work, two extra things must be true:

- The ISA server(s) must trust the server-certificate of the LDAP server. This can be done by adding the public key cert of your issuing CA into the ISA server's Certificate store. Do this by running MMC, adding the "Certificates" snap-in, and choosing "Computer Account". You want to place the cert into "Trusted Root Certification Authorities".
- The server certificate of the LDAP server must meet the criteria specified at the bottom of the following Microsoft article:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ldap/ldap/example_code_for_establishing_a_session_over_ssl.asp

If you have trouble connecting to Secure LDAP, you should confirm whether standard LDAP connections are able to work. If not, then you may have a connectivity problem, your DN parameters might be wrong, or you may not have specified a valid password. Check the ISA Alerts for more information on failing LDAP connections.

Deciding between FBA and Basic

FlexAuth uses several methods to determine whether a client is capable of using an authentication form, or if instead it should just use the basic authentication method.

The first field is a regular expression that specifies one or more user agents.

> **NOTE:** You don't need to understand regular expressions to use FlexAuth. Just click the wizard button to the right of the field and you can enter a simple list using characters and the asterisk (*) as a wildcard.

If a client's User Agent string matches the expression, then FBA will be used. For FBA to work, a client must support an interactive HTML-based logon process, and must be able to accept HTTP Cookies. All recent web browsers can perform these operations, but services such as ActiveSync and clients like Microsoft Office are not able to do so. For this reason, any user agents that are *not* expressly matched by this field will be served with Basic Authentication.

The other setting in this section controls whether FlexAuth treats certain HTTP request methods differently than others. By default, a new session that starts with any request other than “GET”, “HEAD”, or “POST” will be forced to use Basic Authentication **even if the session identifies itself as a browser that matches the User-Agent setting above**. This setting helps recent versions of Microsoft Office to work seamlessly with FlexAuth. You should only disable this setting if it is causing adverse effects (such as browsers that *should* get FBA being told to use Basic instead).

Basic Authentication Settings

In the “authentication popup text” field you can specify a custom string that will appear in the Basic Authentication dialog presented to the user.

Forms Based Authentication Settings

The Internet Domain Name field is extremely important to the proper functionality of the single sign on realm. The suffix you enter here must be shared by **all** servers in this single sign on realm. This is because an HTTP cookie is used to save the session state in the client's browser. This cookie can only be sent to FlexAuth for servers that match the suffix you enter in this field. Any non-matching servers will not be able to participate in the realm. In practice this is not

usually a problem, as most organizations share at least their root domain name across all published servers. If you have more than one suffix, you can still create more than one single-sign-on realm, one for each suffix. Of course the credentials cannot be passed *between* the realms, but within each realm single sign on will work.

The timeout fields are fairly straightforward. FBA users' sessions will be expired after the specified number of minutes of inactivity. If you wish to differentiate between public and private machines, you may select two different intervals. For more on how to choose between public and private machines, see the section on customizing the FBA.

Finally, there is a regular expression that's used to detect logoff requests. (Once again, you can use the wizard to list several simple matches if you don't want to create the regex by hand). When a user's browser requests a URL that matches this field, their session will be invalidated and they will be redirected to the logoff page in the FBA directory (or logoff_basic page, if they were authenticated with Basic Auth). By default, this expression is set to match the OWA logoff URL.

Please note that Basic Authenticated users cannot be reliably "logged off" since the browser stores the credentials and automatically offers them to the server on each request. This is, in fact, one of the biggest reasons why FBA should be used whenever possible!

Additional listener settings

HTTPS Bounce

New in version 1.1. If you publish a web site through both HTTP and HTTPS externally, and want your authentication traffic to be HTTPS but everything else to be HTTP, you can check the "Use HTTPS Bounce" checkbox. In order for this feature to work, you should have one of two possible configurations:

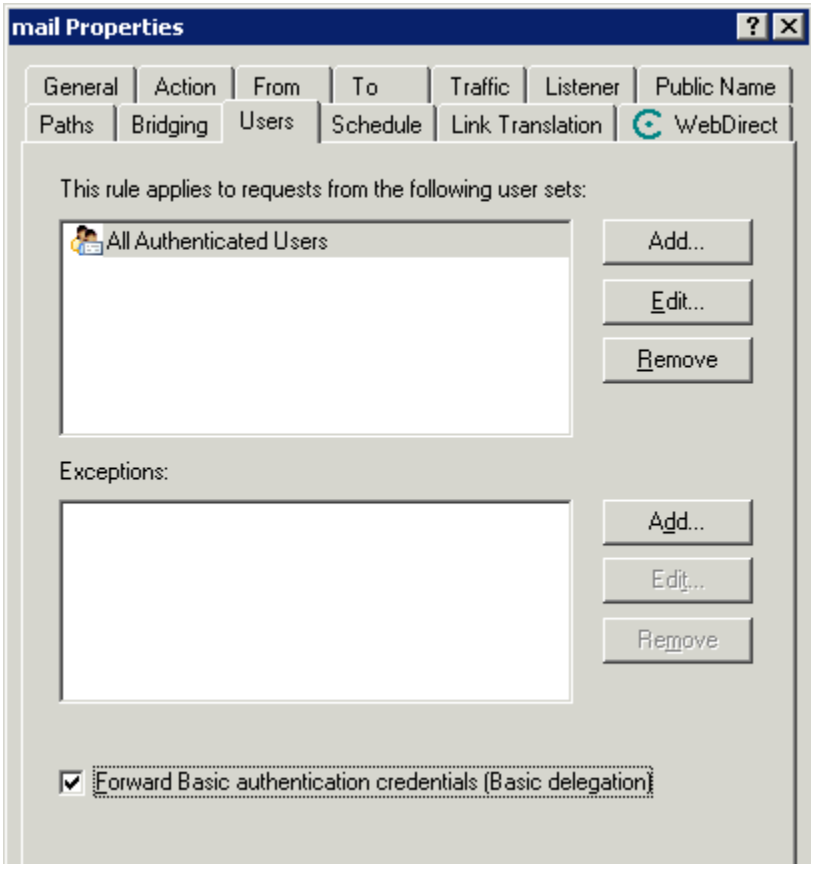
- One FlexAuth-enabled listener that is set to listen on both HTTP and HTTPS, and one publishing rule that uses it.
- Two FlexAuth-enabled listeners, one for HTTP and one for HTTPS, and corresponding publishing rules that use them. The public-name on the two rules must be the same, and make sure that both listeners are members of the same FlexAuth realm!! Also, don't forget to check the HTTPS Bounce box on each listener, or you will get incorrect behavior.

Setting up other listeners

Now that you have completed your realm settings, go into the other listeners that will participate in this single sign on realm and select the appropriate realm from the dropdown.

Configuring your Publishing Rules

Apart from selecting your FlexAuth-enabled Listeners, there is only one part of a Web Publishing Rule that is relevant to FlexAuth: the Users tab.



Forward Basic credentials

If your web server needs to authenticate the user (i.e. if it is OWA or some other service that must know who the user is) then check the “Forward Basic” checkbox on this tab. This will forward all FlexAuth credentials (for Basic and also FBA users) on to the upstream server. For more information on what data is forwarded, see Default Domain.

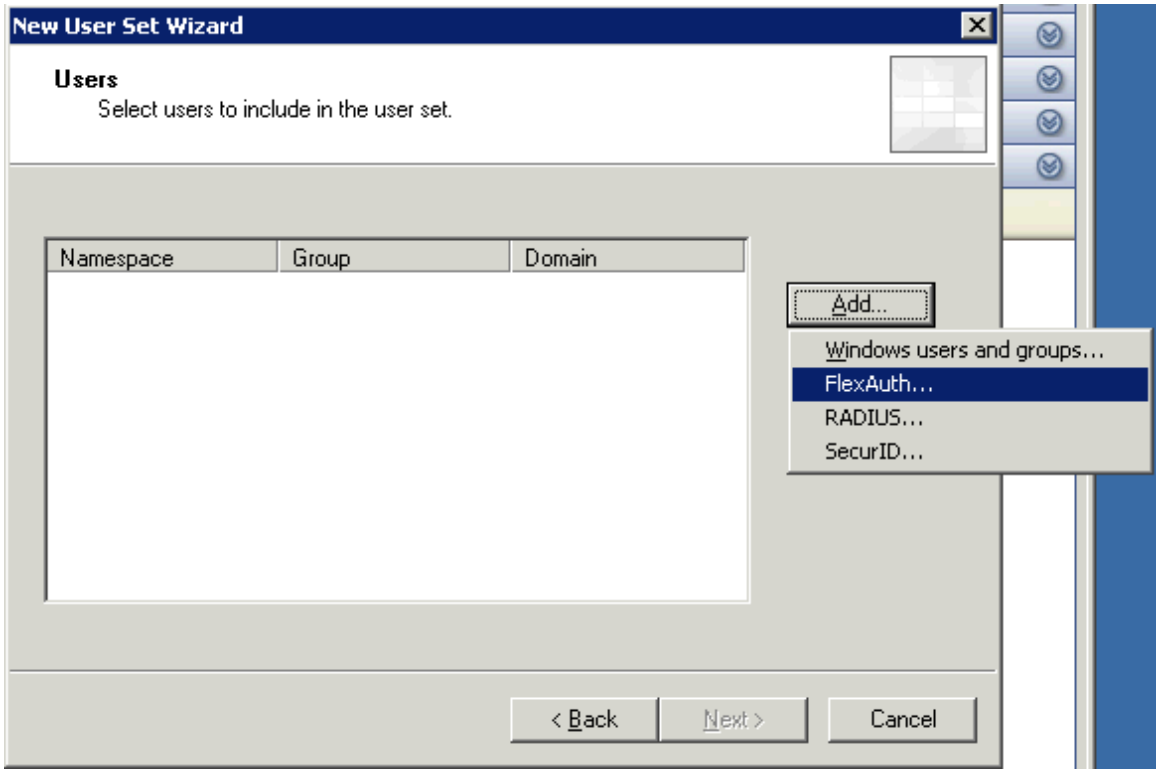
Active Directory (Windows) User Sets

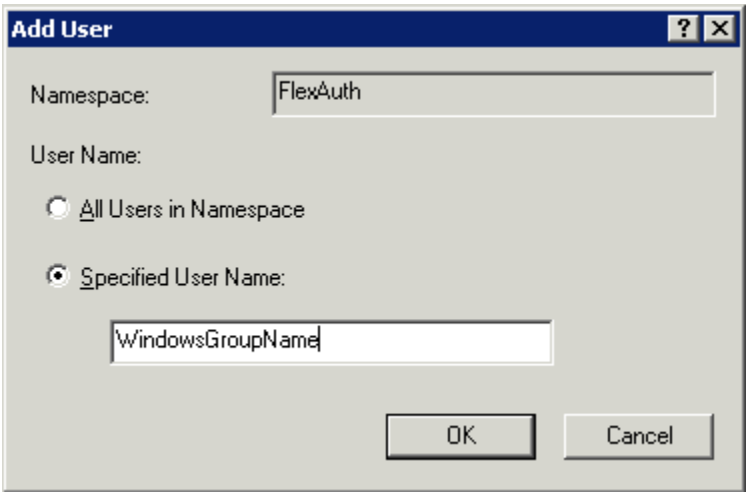
If your realm uses AD Integrated authentication, then your user sets should contain Windows groups and users, as with any standard Publishing Rule.

FlexAuth (LDAP) User Sets

If you are using LDAP authentication, then there is a slight twist added. Since LDAP authentication is not built in to the firewall, ISA does not recognize LDAP users as “Windows” users (furthermore, if you are using LDAP then your ISA

server is not in the domain, so it would have no way to enumerate Windows users and groups at any rate). With LDAP, instead of adding “Windows Users and Groups” you will add “FlexAuth” users and groups.





Despite the poorly named “Specified User Name” field, you may type a Windows Group name *or* a User Name in this field. Build your user sets this way, by specifying “FlexAuth” groups and/or users. Then add these user sets to your Publishing Rule.

Note that if you simply wish to allow all authenticated users instead of locking access down to specific groups or users, you may just use the “All Authenticated

Users” built-in set.

Note that when you use LDAP authentication, the ISA logs will reflect the authenticated users as “FlexAuth\Username” instead of “WindowsDomain\Username”, even though you are still ultimately authenticating against the Windows domain.

Testing your configuration

Now is a good time to stop and make sure everything is set up properly. Apply your changes to the ISA configuration (and then remember to wait until all Array members are synchronized, if you are running Enterprise edition). Now when you direct a web browser to one of your externally published servers, you should receive a simple FBA logon page. If you receive a Basic Auth box or a 401 error instead, go through the Listener's configuration step by step to see if anything has been missed.

Narrowing down problems

If you are using a complex configuration and “something isn't working” it can often be frustrating to find the real cause. To narrow down the source of the problem, you can begin setting items to their simplest states to rule out various portions of the configuration/setup. For example, consider the following simplifying configurations:

- Set the authentication method to Integrated instead of LDAP. LDAP configuration can be tricky, so it's good to find out whether that's your problem. If your server isn't a member of the domain, you can type the NETBIOS name of your ISA server into the “default domain” field, and then define a local user account on that box to test with. This causes FlexAuth to authenticate against the server's local user database.
- Go back to using the example FBA page, instead of your customized one. This helps rule out any bugs that may have been introduced in the customization process.
- Blank out the regular expression used for matching FBA clients. This will cause FlexAuth to use Basic Authentication for all requests. This can be a good way to see if something is fishy with your FBA settings. If everything works fine in Basic, then you know where the problem may lie!

Customizing the FBA page

Support statement

Note: Collective Software support is always happy to help get you started with your customizations. Several example FBA pages will be provided in the installation. Before reporting a bug against the FBA process, please test using

one of the example setups that comes with the filter, instead of your own customized page. This ensures that the problem is not due to a bug in the customization. Collective Software support cannot be responsible for troubleshooting customer-created DHTML and/or code.

Location and types of FBA files

FlexAuth serves all FBA files (DHTML and images) from one fixed directory. If you have installed ISA in the default location, then this directory will be C:\Program Files\Microsoft ISA Server\Collective Software\FlexAuth\HTMLFiles. Please note the following limitations:

- You cannot create subdirectories under this folder, nor can you serve files from other directories on the ISA server (although you could still make absolute HREFs to files hosted on some other server that's not behind the FlexAuth realm).
- The following file extensions are supported: html, htm, jpg, jpeg, gif, png, css, js.
- No ASP or other server-side scripting or processing is supported (client side javascripts will work, however).

These limitations are for security purposes, and the last point is simply due to the fact that ISA is not a full-featured web server that can support server-side scripting technologies.

File Names

The names of the files in the HTMLFiles directory are significant and should not be changed. For example, FlexAuth will always use the file named “login.htm” to serve the FBA form. You may still create other html files (if you wish to serve them via an Iframe or frameset) but the original file names will always be referenced and served by FlexAuth for the login/logoff operations.

Default files

When you first install FlexAuth, a set of default files is created in the HTMLFiles directory. These are:

- login.htm
- logoff.htm
- logoff_basic.htm

You may modify these files and your changes will always be saved (the installer will never delete or overwrite these files once they have been created).

Example files

There are several example files, which begin with “example_”. These files are for example purposes only, and are not used in the filter. **The files will be overwritten and/or removed whenever the installer is run.**

Support files

There is currently one support file, “authenticate.js”. This file is **not** to be modified or customized. **It will be overwritten whenever the installer is run.**

Login page

The login.htm page is an extremely basic form that demonstrates all the options that can be used to communicate with FlexAuth. The form fields used must not be renamed, or else the filter will not be able to read the user's credentials. Any HTML may be used in this file, as long as the form POSTs back to the same page, and the field names are not changed. Furthermore, the “authenticate.js” script must be included at the end of the page, in order to provide facilities to connect the form with FlexAuth properly. If you wish to change the behavior of this script, make a copy and refer to the copy in your login page. **Modifications to the original script will be overwritten by the installer!** Any changes made to the authentication script functionality will be considered “unsupported”. See the comments in the example page for further information.

Note in the example page that there is a checkbox control for the user to assert that they are on a private machine. If you don't wish to allow the user this choice, you can simply hide or remove the control.

There are several options for allowing the user to specify the domain they wish to authenticate against. You may provide a separate form field (a text entry box or dropdown). The user can enter their credentials in the form “username”, “domain\username” or “username@domain.com”. Note that LDAP mode can never search outside of the Base DN, so any domain field entered by the user is not relevant in that case.

Logoff pages

There are two logoff pages used by FlexAuth. One for FBA users “logoff.htm” and one for Basic users “logoff_basic.htm”. Since Basic users cannot be “logged off” reliably until all browsers are closed, this separate file is provided so that the message to the user might be appropriately worded. In addition, there is a client-side script (effective for IE6.1 SP1 and newer **only**) which attempts to clear the browser's memory of Basic credentials.

Note that there is no necessary scripting or logic in the logoff page for FBA users. The session invalidation is performed internally by FlexAuth, and the logoff page just serves as a redirect destination for the user's browser to inform them that they are now logged off. In other words, you can put whatever you want in this

page, as there is no requisite functionality to be maintained.

Supporting files

Images, stylesheets, and javascripts may be stored in the HTMLFiles directory and referred to with relative URLs. Keep in mind that you may not serve files from subdirectories or superdirectories, only the “HTMLFiles” folder itself. If you have a large number of users, keep in mind that serving big graphics files will substantially slow down the loading of the login page. All HTML served by FlexAuth must be processed through the filter thread itself, and caching features are not supported for this content.

Additional Examples

At the time of this document's writing, FlexAuth includes only the one simple set of pages that are used by default. It is expected that one or several more example file sets will be provided in the near future to demonstrate a few more of the customization possibilities. Please see the HTMLFiles directory for the current set of examples. A discussion about DHTML is outside the scope of this document. For complex DHTML programming tasks, we have found the O'Reilly Dynamic HTML reference to be invaluable.

Requests

Thank you for evaluating FlexAuth; we hope it meets the needs of your organization! If you have any feature requests or other comments, please address them to info@collectivesoftware.com.

Appendix A: Regular Expressions

A full discussion of regular expression syntax is beyond the scope of this document. FlexAuth supports perl-compatible extensions to standard regular expressions. All FlexAuth regular expressions are case-insensitive automatically.

Most users will only be interested in very simple expressions. The default expression for matching browser user agents is:

```
MSIE|Gecko
```

Which means match either the string “MSIE” or the string “Gecko”.

In regular expression syntax, a period represents one wild card character. In other words, a period will match exactly one character, but it doesn't matter what that character might be (hence it's a wild card). To get the more customary “as many characters as you want” wild card, you append an asterisk after the period. So:

```
Before.*After
```

would match any string that contains “Before”, followed by 0 or more characters, followed by “After”.

There are many fine tutorials online that go into far more detail about regular expressions. A quick web search will bring up several examples, including

- http://www.regular-expressions.info/
- http://www.cc.gatech.edu/classes/RWL/Projects/citation/Docs/Design/regex.intro.1.doc.html

and many others.

Finally, if you know what perl extensions to regex are, then you clearly don't need for them to be explained in detail here. Of primary interest are:

- non-greedy quantifiers
- look-ahead and behind constructs
- etc.

See http://linuxcommand.org/man_pages/perlrequick1.html for more on perl regex.