
# Configuration for Business Environments

Document Version: 1.4
Software Version: 2.2010.8.1

NOTABLY GOOD LIMITED

# Table of Contents

## *Introduction*

Installing Affixa in your environment is easy. The basic process is to:

- **Configure** – configure Affixa to your requirements;
- **Export** – this will build an MSI installer which you can use to deploy the software and your custom configuration;
- **Deploy**.

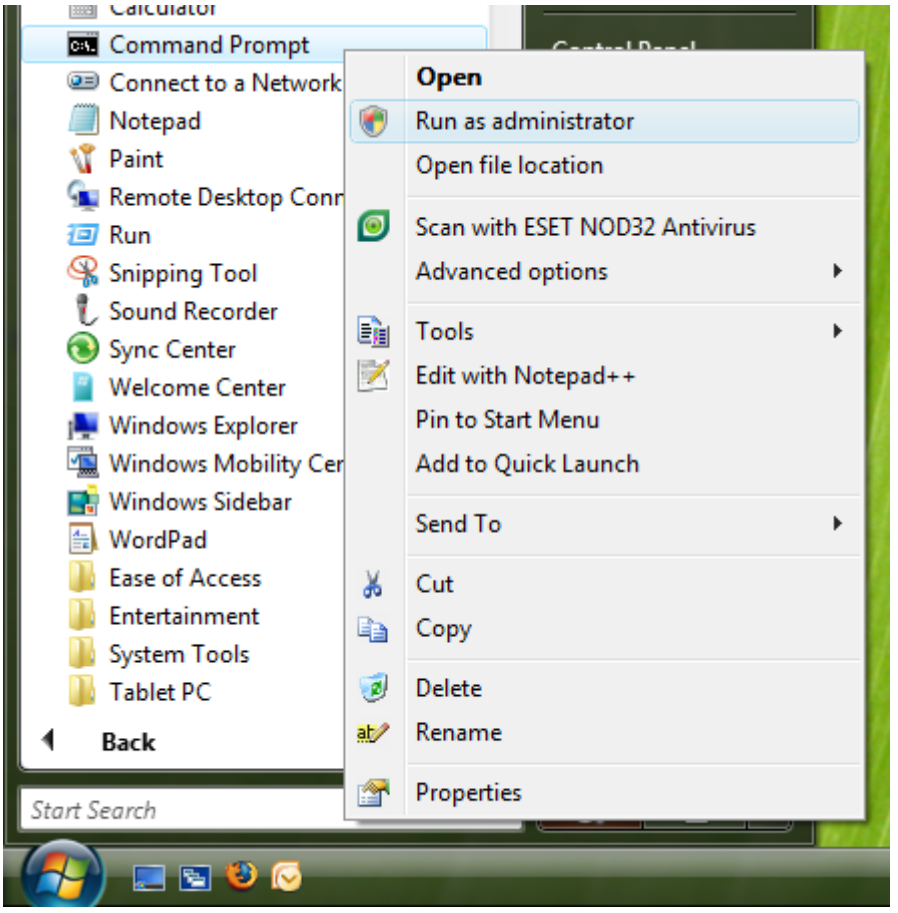
This guide assumes that you are already familiar with the most effective means of deploying MSI installer-based software to your network, and concentrates on the first two stages: configuration and the production of the installer.

## *Accessing the Network Install options in Affixa*

As an administrator:

- Ensure that Affixa is not running;
- Open a new Command Prompt window:

**IMPORTANT:** If you are using Windows Vista, Windows 7 or Windows Server 2008, you must ensure that your Command Prompt window is opened using elevated privileges. You can do this by right-clicking on the Command Prompt option in the Start Menu and choosing "Run as administrator":




At the command prompt, use the "cd" command to go to Affixa's folder on your hard disk, for example:

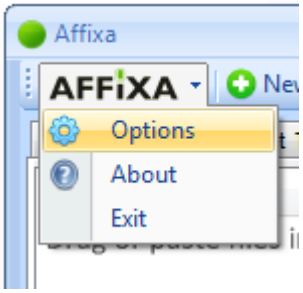
```
cd "\Program Files\Affixa"
```

If you are using a 64-bit machine, the Affixa folder will be found under "Program Files (x86)".

At the command prompt, enter the following command:

```
AffixaTray.exe /NetworkAdmin
```

This  will launch the system tray application. Double-click the Affixa icon and then choose the Options menu item from the Affixa menu:




You'll notice that there's a new tab added to the Options screen on the far-right, called "Network Install". Choose this tab.

# *Configuring Affixa for your Network*

## Email Accounts

For any email account you've configured, Affixa requires you to have given it a custom name. This might be "Bloggs Inc Email", for example.

## Which settings are included in the export?

### Account Settings

When exporting your settings to an MSI Installer, only the "stub" details of your email account will be included by default. In the case of a Google Apps account, this *would* include:

- The Google Apps domain name (e.g. "bloggs.com");
- Any "Preferences" set, including "Launch Gmail directly after creating a message", for example;
- The path to a custom login script, if defined;
- Any "Extras" configured, including the chosen web browser and any automatic recipients;

By default, the following *would not* be included:

- The account username;
- The account password.

When a user first tries using Affixa following installation, they will be prompted for their *own* username and password and this can be recorded for future use.

The exception to all this is if the "Include usernames and passwords in the template" option is chosen on the Configuration screen, as outlined in the following section. This option is NOT recommended.

### General Settings

The choices you have made for the following options will also be included in the template:

- Whether to launch Affixa on start-up;
- Whether to always show the "Choose an Account" screen when sending;
- Attachment shrinking settings;
- Theme settings;
- All Drop.io settings;
- Proxy server configuration.

In addition, Affixa will be set to be the machine's "mailto" handling application.

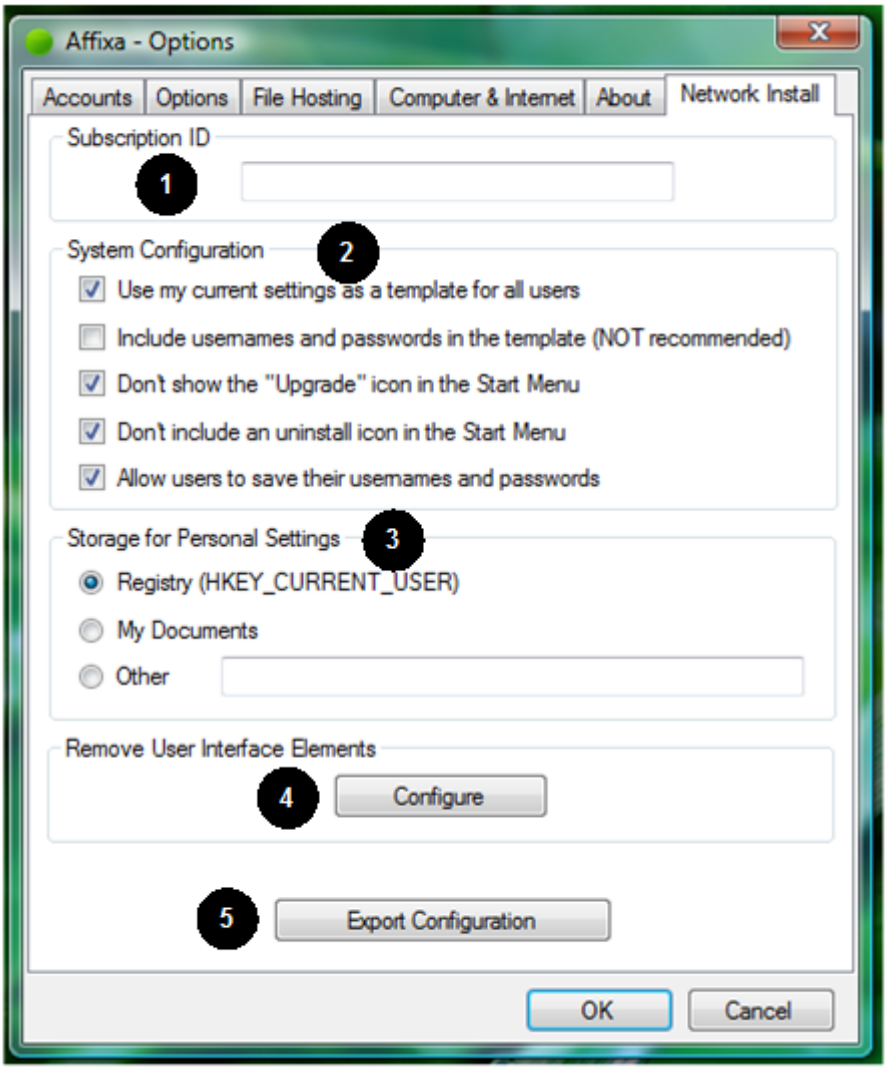
## Single or Custom Sign On

If you have been provided with an XML script for your Single Sign-On or Custom Sign-On environment, this script should be stored in a shared, read-only location.

The login script can be associated with an account on the account's "Preferences" tab. Note that this option is only visible when in Network Install mode.

For more details on custom login scripts, see below.

## Configuration Options




1. **Subscription ID** – this is a mandatory field. This is the subscription code you were emailed upon purchasing your licences.

2. **Configuration** – determines how the Export process should use Affixa's current configuration:

    a. **Use my current settings as a template for all users** – this means all users will use Affixa as per your current configuration. This includes any email accounts that have been configured, *but usernames and passwords will NOT be included unless the following option is checked.*

    b. **Include usernames and passwords in the template** – this option is NOT recommended. This will means what it says – the usernames and passwords you've configured will be included in the settings export and included in your standard build.

c. **Don't show the "Upgrade" icon in the Start Menu** – just a little Start Menu trimming for corporate environments.

d. **Don't include an uninstall icon in the Start Menu** – again, just a little Start Menu trimming.

e. **Allow users to save their usernames and passwords** – if your network security policy does not allow users to save their credentials, you should make sure that this box is not ticked. You should also make sure that the "Remember these details" checkbox is removed from the credentials window (see User Interface Trimming below).

3. **Storage for Personal Settings**

   a. **Registry (HKEY_CURRENT_USER) –** requires Roaming Profiles in situations where users do not always use the same computer: http://technet.microsoft.com/en-us/library/cc784961.aspx

   b. **My Documents** – settings are stored in an encrypted flat file at the location Windows uses for "My Documents" (or "Documents" in Vista, Windows 7 or Windows Server 2008).

   c. **Other** – settings are stored in an encrypted flat file at the location stated. No validation of this location is made by Affixa during configuration.

4. **Remove User Interface Elements –** this allows you to remove certain elements of the UI to present a tailored, streamlined experience for your users, removing the temptation for them to "play" with settings. Further information is available in the following section.

5. **Export Configuration –** this creates the MSI installer and supporting files for you in a location of your choosing.

## User Interface Trimming

This screen allows you to hide elements of the user interface. The "true" or "false" options indicate whether or not an option should be shown: "true" if it should, "false" if it shouldn't.

The options are grouped by theme:

| Group | Description |
|---|---|
| About | The "About" screen shown on the final Options tab and via the "About" option in the Affixa tray application menu. |
| Account Window | The window that pops up when multiple accounts are configured and you choose to send an item using Windows (e.g. Sent To > Mail Recipient) or from an application (e.g. Microsoft Word). |
| Accounts | The Accounts tab in the Options screen. |
| Basket | The items shown for each Basket in the Affixa tray application. |

| | |
|---|---|
| Computer | The “Computer & Internet” tab of the Options screen. |
| Extras | The Extras tab used for each email account. |
| File Host | The File Hosting tab in the Options screen. |
| Gmail | The settings form used by all Gmail and Google Apps email accounts |
| My Computer | The settings form used by local desktop email applications |
| Options | The Options tab of the Options screen. |
| Tab | The tabs shown on the Options screen. This is the quickest way of hiding an entire tab’s content. |
| Tray | The Affixa tray application |
| Yahoo | The settings form used by all Yahoo! email accounts |

## *Network Security*

In some cases, it may violate an organisation’s network security policy for Affixa to save a user’s credentials. In such situations, care should be taken that:

- the option for whether users should be able to save their usernames and passwords is set to “off”;
- the “Remember these details” prompt is hidden using UI trimming.

Using this configuration, each time a user runs Affixa the software will ask for credentials. The credentials will only be used for the session and will then be discarded when Affixa exits.

## *Firewalls and Proxy Servers*

If your organisation restricts internet access in any way, please ensure that HTTPS communication with www.notablygood.com is unblocked at all times. The software will check that the licence being used is valid using a secure callback to this site.

## *Preventing User Updates / Automatic Update Notifications*

In order to prevent users from attempting to install updates or to suppress update notifications, hide the “Check for Updates” button in the “About” section of the UI trimming window.

## *Deploying your customised build of Affixa*

How you deploy your build of Affixa will depend upon your environment.

For Citrix or Terminal Services environments, it should suffice to simply use the installer on your server.

For Windows networks, the best way of deploying Affixa is using Group Policy: http://support.microsoft.com/kb/816102

Ensure that HTTPS access to www.notablygood.com is available at the time of installation.

## *Special considerations for Citrix environments*

In many Citrix environments, an “artificial” stub version of Internet Explorer (iexplore.exe) is deployed. This stub can confuse Affixa when it comes to checking the version of Internet Explorer that is installed on the server.

To avoid this problem, the following registry key must be added:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Affixa]
"CitrixForceIEVersion"="7"
```

This example assumes that the installed version of Internet Explorer is version 7.

## *Internet Explorer – Protected Mode*

Internet Explorer versions 7 and 8 include a “Protected Mode” which is switched on by default in Windows Vista and 7.

If you have configured Affixa to automatically show a draft message after creation, you will need to ensure that *.google.com or *.yahoo.com are in Internet Explorer’s list of sites in the “Trusted Zone”.

To create a policy to add a site to the Trusted Sites security zone:

01. Log on as a member of the Domain Admins group.

02. Open the Active Directory Users and Computers MMC snap-in.

03. Right-click the domain or Organizational Unit where you want to create the GPO and press Properties.

04. Select the Group Policy tab.

05. Press New.

06. Type a name for the new GPO and press Enter.

07. To prevent the policy from being applied to some users or groups, press Properties. Select the Security tab. Add the user or group that you want to prevent from having this policy and clear the Read and the Apply Group Policy boxes in the Allow column. Press OK.

08. Press the Edit button.

09. Navigate through User Configuration / Windows Settings / Internet Explorer Maintenance / Security.

10. Right-click Security Zones and Content Ratings in the right-hand pane and press Properties.

11. Select Import the current security zones and privacy settings. If prompted, press Continue.

12. Press Modify Settings.

13. Select Trusted Sites and press the Sites button.

14. Type *.google.com or *.yahoo.com and press Add.

15. Press Close (or OK) and OK.

16. Press Close (or OK) until all dialog boxes are closed, and close any snap-in windows.

17. Allow sufficient time for the policy to propagate throughout the domain.

# *Custom and Single Sign-On Scripting*

## Introduction

Many business and academic environments are adopting single sign-on systems. This means that Google's own means of logging into Mail accounts is replaced by in-house authentication systems that integrate with Google Apps.

To cope with the wide range of different authentication providers, Affixa has a script-driven login engine. Using this engine, it is possible to:

- Specify a starting URL to load (e.g. the address of your organisation's sign-in screen);
- Automatically complete and post back forms using dynamic and static field values;
- Handle responses from post-backs;
- Validate the outcome of the login process.

## An Example Sign-In Script (based on the Pubcookie system)

```
<?xml version="1.0" encoding="utf-8" ?>
<fba>
  <session>
    <transaction auth="form">
      <url><![CDATA[https://weblogin.university.edu]]></url>
      <form>
        <identify using="name">query</identify>
        <fields>
          <field name="user"><![CDATA[%USER%]]></field>
          <field name="pass"><![CDATA[%PASS%]]></field>
        </fields>
      </form>
    </transaction>
  </session>

  <session>
    <transaction auth="form">
      <url><![CDATA[http://mail.google.com/a/university.edu/]]></url>
      <form>
        <identify using="index">0</identify>
      </form>
      <tests start="0">
        <test type="host" result="1">mail.google.com</test>
      </tests>
    </transaction>
  </session>

  <session>
    <transaction auth="none">
      <url><![CDATA[http://mail.google.com/a/university.edu/h/?v=b&pv=tl&cs=b]]></url>
    </transaction>
  </session>
</fba>
```

## Explanation of Script

A login attempt is broken down into one-or-more sessions, each containing one-or-more transactions. A session can be thought of as a sequence of connected transactions. Transactions that are not connected exist in separate sessions.

The first session handles logging in to the University's sign-on portal. Its first transaction connects to the portal at the web address https://weblogin.university.edu . The returned web page contains a form. A web page can contain many forms, so the specific form used for authentication is identified using the "name" attribute. This should equal "query" in order to identify the following HTML <form> tag, for example:

```
<form name="query" method="POST" action="https://weblogin.university.edu/"
enctype="application/x-www-form-urlencoded" autocomplete="off">
```

All the default and hidden values of the form are then loaded from the web page. These defaults and hidden values can be overridden by the different <field> elements in the script. In this example, we override the default values of HTML <input>s with the names of "user" and "pass". For each of these, the user's mail account username and password are pulled in dynamically.

The form is then submitted back to the sign-on portal and, if all is well, the portal returns a web page confirming that the user has logged in successfully.

The second session takes care of logging in to Google Apps Mail. The script tells the engine to connect to http://mail.google.com/a/university.edu/ and to expect a form in return. In reality, this form comes after several HTTP redirects between Google Apps and the University's sign-on server and is used to pass encrypted values back to Google via a POST request. If you were using a normal web browser such as Mozilla Firefox, you probably wouldn't be aware that this form existed as it ordinarily submits itself automatically using Javascript.

This time, the form is identified by its index in the web page. The index is zero-based, so the first form will have an index of 0, the second will have an index of 1, etc.

We don't care about changing any of the form's values so there are no <field> elements this time. The form is simply loaded with its default values and is then submitted.

After completing the second session, a test is performed to ensure what has happened in the session is correct. This is a simple test to check that the hostname of the final URL contains "mail.google.com". If it doesn't (e.g. if it contains "weblogin.university.edu"), we know that something went wrong somewhere. This particular set of tests starts off with the assumption that the test fails (start="0") until it is proven successful by an individual test (result="1" when the hostname contains "mail.google.com").

The final session directs the engine to Google Mail's compose page. This is a necessary part of all scripts and should be present without fail.

## Description of Script Elements

### Root Element

- The root element can be of any name. In the example above, “fba” is used for “forms based authentication”.

### session

- A session element is a container for one-or-more transactions and has no attributes of its own.

### transaction

- Can exist as a child of a session element, or as the child of another transaction element.
- **auth** attribute (required) - can have the following values:
    - **none** – if no kind of authentication or forms-processing is required.
    - **form** – if the response to the request should be processed as an HTML form. This requires the presence of a “form” sub-element.
    - **ntlm** – this will include the user’s current Windows credentials in the headers of the request using integrated-Windows authentication.
    - **basic** – this will send the user’s mail account username and password in the headers of the request using basic authentication.
    - **digest** – this will send the user’s mail account username and password in the headers of the request using digest authentication.
- **url** sub-element:
    - optional, however the first transaction in each session should have this attribute set. CDATA sections should always be used to encapsulate the address.
    - Where no url is given, the input of the transaction is deemed to be the output of the previous transaction.

### form

- Acts as a container for other form-related elements.
- Is required when the transaction’s “auth” attribute is set to “form”.
- Is the child of a transaction and the parent of “identify” and “fields” elements.

### identify

- **using** attribute (required) – can have the following values:

    - **index –** finds the required form by zero-based index.
    - **name –** finds the required form by the HTML <form> tag's name attribute.
    - **id –** finds the required form the HTML <form> tag's ID attribute.
- The value of this element is either a number (in the case of "index"), or a string (for "name" or "id").

## fields

- Optional element, but required to contain "field" sub-elements.
- Is the child of a form element.

## field

- **name** attribute (required) – indicates the name of the HTML <input> that the contained value should be applied to.
- The following dynamic values are available:
    - **%USER%** - this fills the form field with the current user's mail username;
    - **%PASS%** - this fills the form field with the current user's mail password.
- Static values can also be given, e.g. <![CDATA[My Static Value]]>
- If you do not wish a checkbox value to be returned to the server as checked, use <![CDATA[]]> as the field's value.
- The value should always be contained within a CDATA section.

## tests

- Optional element, but required to contain "test" sub-elements.
- **start** attribute (required) – indicates what the starting state of the tests should be. If you want to assume the test fails unless proven otherwise, the start value should be "0" (i.e. false). If you want to assume that the test passes until proven otherwise, the start value should be "1" (i.e. true).

## test

- **type** attribute (required) – value can be:
    - **host –** checks whether the final URI's hostname contains the indicated value (NB: this is not an exact match but a substring search).
    - **path –** checks whether the final URI's path contains the indicated value (NB: this is not an exact match but a substring search).
- **result** attribute (required) – value can be:
    - **0 –** if the test passes, return "false". If the test fails, no value is returned.
    - **1 –** if the test passes, return "true". If the test fails, no value is returned.

## Script Authoring Service

Notably Good Ltd offers a bespoke script-authoring service on a time and materials cost basis. Please contact support@notablygood.com for further details, providing details of your sign-on environment.

## *Support*

Free product support is available via email for any configuration queries. Please contact support@notablygood.com with any questions and we will do our best to respond within 48 hours.