

# AD2SIGNATURE

AD2Signature takes personalised information stored in Active Directory and places it in a predefined template which gets stored in the users Outlook area of the profile the user is using on that computer.

*FreeMEG Software  
2011*

## Contents

Rationale .....	2
System Requirements .....	3
Quick HOW-TO Guide .....	3
The Command Line .....	5
Default Values .....	7
Appendix A: Common LDAP Attributes used in Active Directory .....	7
Appendix 2: Known Issues .....	8
RTF (Rich Text File) – Error upon rewriting .....	8
Solution .....	9
Appendix 3: Software License .....	9
Updates .....	10

# AD2Signature

*A complete solution for site wide standardised Outlook Signatures*

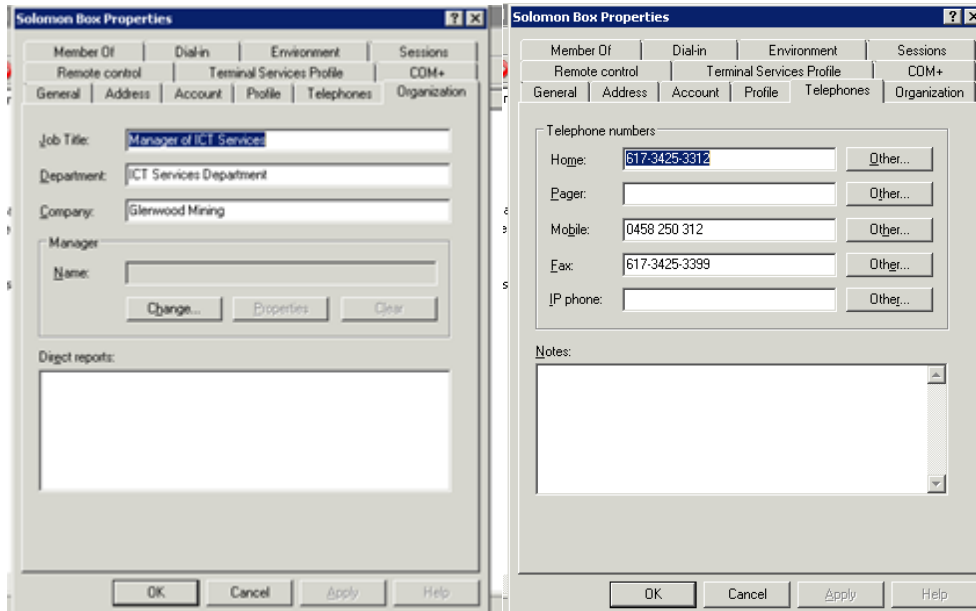
**AD2Signature takes personalised information stored in Active Directory and places it in a predefined template which gets stored in the users Outlook area of the profile the user is using on that computer.**

## Rationale

So you're in a situation where you need to standardise the Outlook Signatures for your organisation. You need to make it simple and easy to change when required. You need to be able to make it as easy as possible for the end user, to the point where they may not even know this program is running and you wouldn't mind it being a little bit simple for yourself.

The problem with site wide email signatures, are people have different roles, phone numbers, contact details. You may have roaming, or local profiles. People may be changing their roles, offices, locations and to top that off, there may not necessarily be one global signature and various departments may have their own stationary or styles.

**Enter AD2Signature!** As you may already know, Active Directory stores information about each user. Take a look at the screen shots from **Microsoft's Active Directory Users and Computers** program:



**So why not use this to your advantage?**

With AD2Signature it is a simple process:

- 1) Create your signature template using Microsoft Outlook
- 2) Using AD2Signature, extract this template to a shared location

- 3) In Group Policy, Login Scripts or an Icon on the desktop, run AD2Signature to create/standardise the Outlook Profile for that user.

## System Requirements

AD2Signature has been tested on Windows 7 64-bit with Outlook 2010 on a Windows Small Business Server 2008

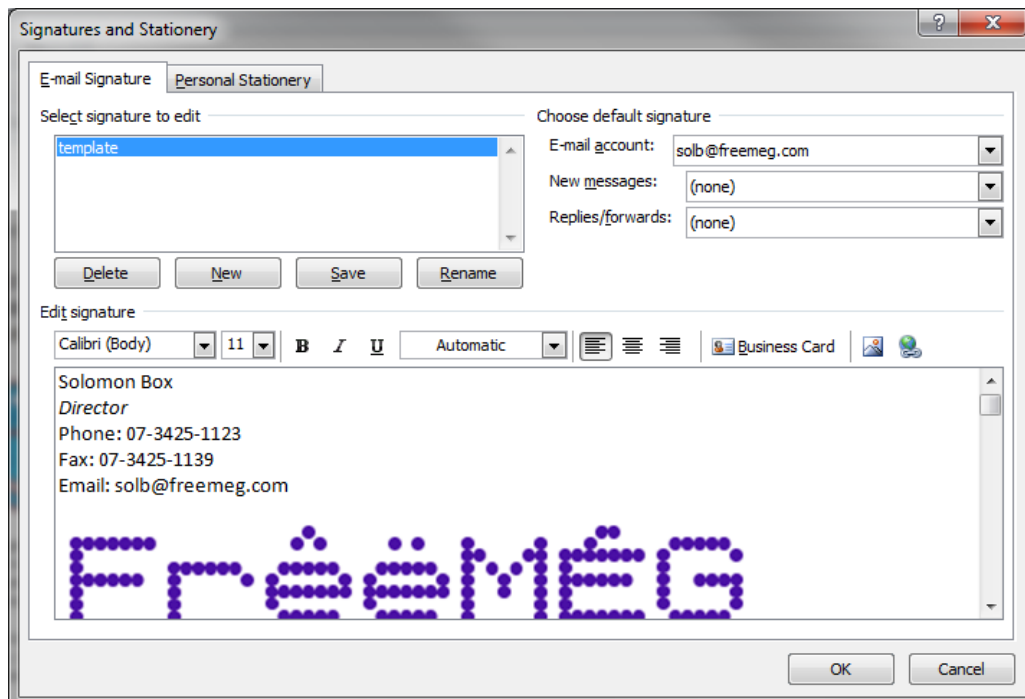
However we believe that the software will work on:

- Any computer running Windows 2000 or better
- Office 2003 or better
- A computer with a Pentium 4 or better processor
- A active directory forest from Windows Server 2000 or better

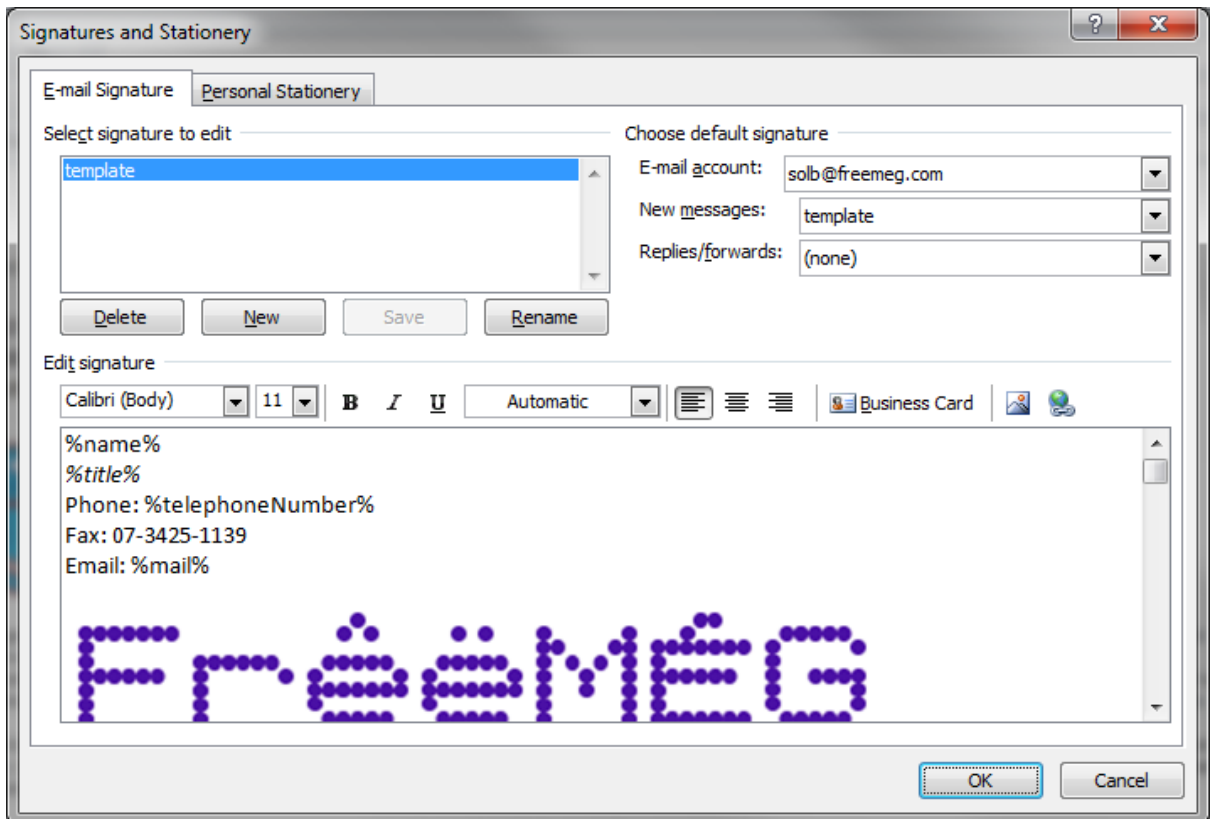
## Quick HOW-TO Guide

- 1) Create your shared location on the server. AD2Signature can run directly from a UNC file share (hidden or not) or a mapped drive.
- 2) Copy AD2Signature.exe into that location
- 3) Run outlook and create an email signature. Call it **template** when it asks for a name.
- 4) When editing signature, simply replace the fixed text with **%ldapattribute%** variables. Then save your signature.

**Confused? So if you have an email signature that looks like this:**



You would replace the information that varies with your LDAP Attributes from active directory and it would end up looking something more like this:



For a listing of common LDAP Attributes refer to the appendix at the end of this document.

- 5) On your workstation from the Command Prompt run AD2Signature from its shared location on the network with the extract parameter.

**Example: \\trillian\apps\AD2Signature\AD2Signature.exe /EXTRACT**

AD2Signature will now extract the required signature files from your local machine and place them in a subdirectory called template in the shared location on the server.

Using the above example: \\trillian\apps\AD2Signature\template

- 6) On your clients workstations, somehow run the following command:

**\\trillian\apps\AD2Signature\AD2Signature.exe /APPLY**

This will copy down the signature files from the server to the users profile. It will overwrite any signature that is already there with the same name. It will then rewrite the template by replacing the %ldattribute% variables with information from Active Directory. Leaving a signature with the users customised contact information. It will then ensure that this signature becomes the default signature for the user on the machine when they go to write new email.

In this configuration users can still edit their signature and it will stay in effect until the next

time AD2Signature is run. If you want to prevent users from editing their signature, just add the /FORCE parameter to the command line:

```
\\trillian\apps\AD2Signature\AD2Signature /APPLY /FORCE
```

Want the signature to apply to New Messages and Replies? Add the /REPLY parameter to the command line:

```
\\trillian\apps\AD2Signature\AD2Signature /APPLY /REPLY
```

And that is that... Add this command line to your Group Policy, login Script, Startup Program Group in the Start menu, Registry or even as an icon on the desktop for your user to run when required.

**But that is not just it - AD2Signature is highly configurable below is further information you may find useful**

## The Command Line

AD2Signature has many command line options that can be added to change the behaviour of the program.

Below is a list:

**/PROFILE:<name>** - By default when Outlook is first run on a computer it creates a profile by the name of **OUTLOOK** on the system. You can verify this by going to Windows Control Panel -> Mail -> Show Profiles. In some organisations, this profile name is customised. By adding /PROFILE:<name> to the command line, AD2Signature will look for and make changes to a differently named outlook profile instead. **Example AD2Signature /APPLY /PROFILE:freemeg**

**/DELETE:<name>** - This will delete the signature with <name> from the users outlook profile and reset outlook to have no signature. If you do not specify a name, AD2Signature will delete the profile named **template**. **Example: AD2Signature /DELETE:marketing**

**/PATH:<name>** - This parameter tells AD2Signature where to find the signature files if they are not located in the same directory as AD2Signature. This parameter can also be used with the **/EXTRACT** parameter to specify where to put the signature files that are extracted from the users profile when creating the template. Some scenarios where this may be useful include, when you have local servers on a large server VPN and you want the program to refer locally. When you have multiple signature templates and you want to **apply one that is not named template**.

Example when applying a signature to a user's profile:

```
AD2Signature /PATH:\\trillian\shared\signatures\facilities /APPLY /FORCE
```

Example when extracting signature from a user's profile, when creating the template:

```
AD2Signature /EXTRACT:accounts /PATH:\\trillian\shared\signatures\accounts
```

**/DOMAIN:<name>** - by default, AD2Signature will query the local workstation to figure out what domain it is a part of, and then connect to one of the domain controllers to grab the LDAP Attributes that are listed in the signature template. In some configurations, AD2Signature will get it wrong. This parameter will force AD2Signature to try and connect to a particular domain only. **Example:**

**AD2Signature /DOMAIN:freemeg.local /APPLY /FORCE**

**/SEARCH:<name>** - When AD2Signature rewrites the LDAP Attributes in the signature file with real data from Active Directory it determines the username of the currently logged on user and then searches Active Directory for a match using the standard **sAMAccountName** attribute.

**sAMAccountName** traditionally is the username in active directory. In some organisations the current user may match up to a different attribute in active directory. The attribute **cn** is another popular choice. Either way, by specifying the attribute name in the parameter, AD2Signature will use this instead of the standard. **Example: AD2Signature /SEARCH:cn /APPLY**

**/APPLY** – When this parameter is added to the command line, AD2Signature will modify Outlook to by default use this signature template for all new emails. If this parameter is not present on the command line, Outlook will have the signature template added. But the user will be left with the choice of whether they want to use the signature or not. **Example: AD2Signature /APPLY**

**/REPLY** – When this parameter is added to the command line, AD2Signature will modify Outlook to by default use this signature template when replying to email. If this parameter is not present on the command line, Outlook will be untouched. **This parameter must be used with the /APPLY parameter. Otherwise it will be ignored. Example: AD2Signature /APPLY /REPLY**

**/FORCE** – When this parameter is added to the command line, AD2Signature will modify outlook so that the end user can no longer freely edit their signature once it is applied to the user profile. They also will not be able to select other signatures as the whole signature section will be disabled. **This parameter must be used with the /APPLY parameter, otherwise it will be ignored. Example: AD2Signature /APPLY /REPLY /FORCE**

**/EXTRACT:<name>** - This parameter cannot be used with any other parameter except for the **/PATH** parameter. This will extract the signature template of name from the current user's profile and place it in the shared location either specified by the **/PATH** parameter or a subfolder in the same location of where **AD2Signature.exe** is located. If **<name>** is not specified, the **/EXTRACT** parameter will assume you are extracting the signature named **template**.

**Example: AD2Signature /EXTRACT**

**Example: AD2Signature /EXTRACT:accounts /PATH: \\trillian\shared\signatures\accounts**

**/VALUE-<name>:<value>** - This is a very special parameter. It is used to override an LDAP Attribute value that is stored in Active Directory by specifying a different value at the command line. This can be useful in a scenario where a user is at a different workstation to where they usually logon and the phone numbers, or contact details are different to the information stored in active directory. This can also be used instead of relying on active directory for information. The **<name>** is replaced with the name of the Attribute you want to override. The **<value>** is the value that should appear in the signature template instead of the LDAP Attribute Variable.

**Example: (A user is working in an office where the phone number is 33220332 instead of their usual 33220312 recorded in active directory)**

**AD2Signature /APPLY /FORCE /REPLY /VALUE-telephoneNumber:33220332**

**Example: (A user is logged on with a generic user account with Casual for a name instead of theirs. ICT Services has setup an icon with her full name instead)**

**AD2Signature /APPLY /FORCE /REPLY /VALUE-name:"Veronica Lathom"**

## Default Values

In some situations when setting up a site wide template there may not always be information in active directory for a particular user and a default value may be assumed. For example a organisation may have a main telephone switch number and private numbers for certain people. Those with private numbers will have them listed in active directory, where as everyone else just lists the generic phone number in their email signature.

AD2Signature caters for scenarios like these by providing a **AD2Signature.ini** file with a **[DefaultValues]** header. Default values can be setup in this file for specified LDAP Attributes. In the above scenario the following entry could be placed into the ini file:

*[DefaultValues]*

*telephoneNumber=32908900*

## Appendix A: Common LDAP Attributes used in Active Directory

Below is a list of common attribute names and what they are referring to in Active Directory:

LDAP Attribute Name	Definition
<b>c</b>	Country or Region
<b>CN - Common Name</b>	CN=Guy Thomas. Actually, this LDAP attribute is made up from givenName joined to SN.
<b>company</b>	Company or organization name
<b>department</b>	Useful category to fill in and use for filtering
<b>description</b>	What you see in Active Directory Users and Computers. Not to be confused with displayName on the Users property sheet.
<b>displayName</b>	displayName = Guy Thomas. If you script this property, be sure you understand which field you are configuring. DisplayName can be confused with CN or description
<b>givenName</b>	Firstname also called Christian name
<b>homephone</b>	Home Phone number, (Lots more phone LDAPs)
<b>I (Lower case L)</b>	L = Location. City ( Maybe Office
<b>location</b>	Important, particularly for printers.
<b>mail</b>	An easy, but important attribute. A simple SMTP address is all that is required billyn@ourdom.com
<b>manager</b>	Boss, manager



<b>mobile</b>	Mobile Phone number
<b>name</b>	name = Guy Thomas. Exactly the same as CN.
<b>ObjectClass</b>	Usually, User, or Computer
<b>OU</b>	Organizational unit. See also DN
<b>postalCode</b>	Zip or post code
<b>sAMAccountName</b>	sAMAccountName = guyt. Logon name, must be unique in the domain. Can be confused with CN.
<b>SN</b>	SN = Thomas. This would be referred to as last name or surname.
<b>st</b>	State, Province or County
<b>streetAddress</b>	First line of address
<b>telephoneNumber</b>	Office Phone

## Appendix 2: Known Issues

### RTF (Rich Text File) – Error upon rewriting

AD2Signature can terminate with error when rewriting the RTF File. You may see an error similar to the following screen dump:

*FreeMEG's AD to Signature - Outlook Signature Updater  
Version 1.0 (c) 2011 - <http://www.freemeg.com/>*

*Using Outlook 14*

*Connecting to FREEMEG*

*Current user is solb*

*Creating signature files in C:\Users\solb\AppData\Roaming\microsoft\signatures\template*

*ReWriting template.htm*

*ReWriting template.rtf*

***ERROR: Access violation at address 004ED1E0 in module 'AD2Signature.exe'. Read of address 05A04000***

This occurs because of formatting issues within the RTF document itself. This can easily occur if the formatting changes between the characters that make up a LDAP attribute variable eg **%mail%**. When the RTF file is saved it does not cleanly save the variable in its entirety. When the rewrite tries to read the attribute name it runs out of memory.

When the RTF file is opened in a program like NotePad++ you can see the issue quite easily. Below is a screen shot of what the RTF file should store for the ldap attribute variable. See how the **%name%** is fully intact.

The screenshot shows a Notepad++ window titled "L:\projects\AD2Signature\template\template.rtf - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations and editing. The text area shows the following RTF code:

```

43 \newtblstyru\snogrowautofit\usenormstyforlist\noindnmbrts\felnbrelev\nocxsptable\indrlsweleven\noafcnst
44 \ilfomacatclnup0\ltrpar \sectd \ltrsect\linex0\endnhere\sectdefaultcl\sftnbj {\*\pnseclv11\pnucrm\pnstar
45 {\pntxta .}}{\*\pnseclv14\pnlcltr\pnstart1\pnindent720\pnhang {\pntxta }}{\*\pnseclv15\pndec\pnstart1\p
46 \pnlcrmm\pnstart1\pnindent720\pnhang {\pntxtb ({\pntxta })}}{\*\pnseclv18\pnlcltr\pnstart1\pnindent720\pn
47 \ql \li0\ri0\widctlpar\wrapdefault\aspalpha\aspnum\faauto\adjustright\rin0\lin0\itap0\pararsid12401163 \
48 \rtlch\fcs1 \af31506 \ltrch\fcs0 \insrsid1714557 \hich\af31506\dbch\af31505\loch\f31506 %name%}{\rtlch\f
49 \rtlch\fcs1 \af31506 \ltrch\fcs0 \i\insrsid1714557 \line }}{\rtlch\fcs1 \af31506 \ltrch\fcs0 \insrsid1714
50 %telephoneNumber%
51 \par \hich\af31506\dbch\af31505\loch\f31506 Fax: 07-3425-1139}{\rtlch\fcs1 \af31506 \ltrch\fcs0 \insrsid
52
53
54 \par }}{\rtlch\fcs1 \af31507 \ltrch\fcs0 \insrsid1714557 \line }}{\rtlch\fcs1 \af31507 \ltrch\fcs0 \i\fs16
54 {\sp{\sn fFlipV}{\sv 0}}{\sp{\sn pibFlags}{\sv 2}}{\sp{\sn fLine}{\sv 0}}{\sp{\sn fLayoutInCell}{\sv 1}}

```

The status bar at the bottom indicates: Normal text file, length: 106756, lines: 744, Ln: 48, Col: 89, Sel: 7, Dos\Windows, ANSI, INS.

Now let's look at a broken LDAP Attribute variable in a RTF file that causes a crash:

The screenshot shows a Notepad++ window titled "L:\projects\AD2Signature\template\template.rtf - Notepad++". The menu bar and toolbar are the same as in the previous screenshot. The text area shows the following RTF code:

```

43 nbrelev\nocxsptable\indrlsweleven\noafcnsttbl\afelev\utin1\hwelev\sp\tpgpar\notevasp\notrkrconstfrctbl\nc
44 faulclcl\sftnbj {\*\pnseclv11\pnucrm\pnstart1\pnindent720\pnhang {\pntxta .}}{\*\pnseclv12\pnucrltr\pnstar
45 {\pntxta }}{\*\pnseclv15\pndec\pnstart1\pnindent720\pnhang {\pntxtb ({\pntxta })}}{\*\pnseclv16\pnlcltr
46 \pnseclv18\pnlcltr\pnstart1\pnindent720\pnhang {\pntxtb ({\pntxta })}}{\*\pnseclv19\pnlcrmm\pnstart1\pnin
47 stright\rin0\lin0\itap0\pararsid12401163 \rtlch\fcs1 \af31507\afs22\alang1025 \ltrch\fcs0 \fs22\lang3081
48 \dbch\af31505\loch\f31506 %name%}{\rtlch\fcs1 \af31506 \ltrch\fcs0 \insrsid1714557 \line }}{\rtlch\fcs1
49 \lch\fcs1 \af31506 \ltrch\fcs0 \insrsid1714557 \hich\af31506\dbch\af31505\loch\f31506 Phone: }}{\rtlch\fcs
50
51 }}{\rtlch\fcs1 \af31506 \ltrch\fcs0 \insrsid1714557 \line \hich\af31506\dbch\af31505\loch\f31506 Email: }}
52
53 }}{\rtlch\fcs1 \af31507 \ltrch\fcs0 \i\fs16\insrsid1714557 {\*\shppict{\pict{\*\picprop\shplid1025{\sp{\s
54 ne}{\sv 0}}{\sp{\sn fLayoutInCell}{\sv 1}}}}{\picscalex100\picscaley100\piccrop10\piccropr0\piccropt0\picc

```

The status bar at the bottom indicates: Normal text file, length: 106756, lines: 744, Ln: 48, Col: 89, Sel: 65, Dos\Windows, ANSI, INS.

See how there is a large amount of formatting data between the variable name. To AD2Signature instead of seeing %name% it sees %name%}{\rtlch\fcs1 \af31506 \ltrch\fcs0 \insrsid1714557 \line }% AD2Signature cannot possible resolve this variable to data in an ldap attribute but attempts to anyway. It eventually errors.

### Solution

Open the original template and remove the entire variable. Then retype the whole variable in its entirety with no formatting or changes that occur in between the beginning and end of the variable. Once saved it should save the RTF with the variable name intact as per the first screen shot above.

## Appendix 3: Software License

The AD2Signature program is licensed as freeware by FreeMEG software <http://www.freemeg.com/>

The license allows you to use this unrestrictive throughout your organisation. This does not give you the right to distribute the software to other sites. You may however link to the FreeMEG Software website.

Basic Support is provided via forums through the FreeMEG Software website, but the response time is at our discretion. That is if we choose to respond at all. If you want modifications to the software to be made then FreeMEG Software reserves the right to quote for that work to be done.

FreeMEG Software accepts no liability for the loss of data from any version of the AD2Signature, including upgrades and cross grades. FreeMEG Software accepts no responsibility for any loss of data or infrastructure on your site.

FreeMEG Software reserves the right to terminate free licensing without further notice. FreeMEG Software reserves the right to terminate anyone's license for the AD2signature Software.

All intellectual property for the AD2Signature Software (unless otherwise acknowledged), remains in the ownership of FreeMEG Software and its founder Solomon Box.

## Updates

Updates can always be found on our website:

<http://www.freemeg.com/>