



Hawaii Green IT

The 1E Web WakeUp Guide

Remote machine wake up over a web link

The 1E Web WakeUp Guide

Version 1.5 document revision 14

© 1E Ltd 2009

All rights reserved. No part of this document or of the software ("the software") to which it relates shall be reproduced, adapted, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without permission from 1E Ltd. It is the responsibility of the user to use the software in accordance with this document and 1E Ltd shall not be responsible if the user fails to do so. Although every precaution has been taken in the preparation of this document, 1E Ltd and the authors assume no responsibility for errors or omissions, nor shall they be liable for damages resulting from any information in it.

Trade marks

1E is a trade mark of 1E Ltd registered in the UK[, EU and US], with registration applied for in Australia. The 1E device is a trade mark of 1E Ltd registered in the UK and EU, with registration applied for in the US and Australia. MICROSOFT, WINDOWS NT, WINDOWS 2000, WINDOWS XP are all trademarks of Microsoft Corporation in the United States and other countries.

Contents

Section 1	Introduction	1
1.1	What will 1E Web WakeUp do for me?	1
1.2	What is new in 1E Web WakeUp?.....	1
1.3	Conventions used in this guide	1
	Cross references	1
	Notes	1
	Code fragments and command-lines	1
1.4	Who is this guide for?	1
Section 2	Requirements	2
2.1	1E Web WakeUp client requirements	2
2.2	Server requirements.....	2
2.3	1E Web WakeUp service account	2
	Basic 1E Web WakeUp service account permissions	2
	1E Web WakeUp service account DCOM permissions.....	2
	1E Web WakeUp service account WMI permissions	3
2.4	1E Web WakeUp Installer account.....	3
	Basic permissions.....	3
	WMI permissions.....	4
Section 3	1E Web WakeUp Overview	5
3.1	1E Web WakeUp Architecture	5
	The wake up flow through the 1E Web WakeUp architecture	5
Section 4	Installing 1E Web WakeUp	7
4.1	Installing the 1E Web WakeUp Server using the installation wizard	7
4.2	Post Installation Configuration.....	7
Section 5	Configuring 1E Web WakeUp Security	8
5.1	Enabling 1E Web WakeUp security	8
5.2	Adding users to the 1E Web WakeUp security database.....	8
5.3	Securing the 1E Web WakeUp Authorization web service	8
	Locking down security.....	8
Section 6	Using 1E Web WakeUp	9
6.1	Logging on to 1E Web WakeUp.....	9
6.2	Installing the Web WakeUp ActiveX control	9
6.3	The 1E Web WakeUp home page	10
6.4	Registering a machine for Wake Up.....	10
6.5	Waking up a registered machine.....	11
	The registered machine is already awake	12
	The registered machine is not awake.....	13
6.6	Waking a machine by name	15
	Incorrect wake up names	15
6.7	Waking a machine using search.....	18

	Domain and machine name search.....	18
	Search result limits.....	18
	An example search.....	19
	A wake up timeout.....	22
6.8	The machine status icon.....	23
	The machine wake up progress icon.....	24
Section 7 Troubleshooting		25
7.1	Checking the problem.....	25
7.2	Environmental issues.....	25
7.3	Contacting 1E support.....	25
	Creating a technical report.....	25
Section 8 Further Information		26
8.1	Contact details.....	26
	Website.....	26
	Telephone and fax.....	26
	Post.....	26
	Sales.....	26
	Technical support.....	26
Appendix A The 1E Web WakeUp API		27
A.1	Using the 1E Web WakeUp API.....	27
	Creating a service reference.....	27
	Creating a web reference.....	27
A.2	The 1E Web WakeUp API Method Reference.....	28
	AddAuthorisedMachine.....	28
	ClearRegisteredMachine.....	28
	DeleteAuthorisedMachine.....	29
	FindMachineExact.....	29
	FindMachines.....	30
	FindRegisteredMachine.....	30
	GetAuthorisedMachines.....	31
	GetMaxItemCount.....	31
	Ping.....	32
	RegisterMachine.....	32
	UpdateAuthorisedMachine.....	33
	WakeByName.....	33
	WakeMachines.....	34
Appendix B Configuration Options		35
B.1	Implementing changes.....	35
B.2	WakeUp Web Services options.....	35
	Using SMS/ConfigMgr as the source machine database.....	35
	Setting the maximum number of machines returned.....	36

	Turning authorization on or off	36
	Setting the size of the machine search buffers	37
B.3	Web options	38
	Turning the use of the ActiveX control on or off	38
Appendix C Automatically setting the authorization table		39
C.1	The authorization input file format	39
C.2	Setting the authorization input file	39
C.3	When the SQL Server Agent job is run.....	40

Section 1 Introduction

1E Web WakeUp enables the wakeup of specific machines via a web site. It is primarily aimed at the end user who needs to access their work machine outside office hours from a remote location. 1E Web WakeUp integrates with the 1E Agility Framework and SMS/ConfigMgr to provide machine search and status capabilities.

1.1 What will 1E Web WakeUp do for me?

1E Web WakeUp enables machines to be woken up outside office hours and off premises thereby enabling them to be shut down when not in use. By enabling the waking of registered machines it provides a simple interface to ensure that even non-technical users can get their work machines up and running when they are needed.

1E Web WakeUp also provides an API that enables its wake up capabilities to be used by 3rd party applications.

1.2 What is new in 1E Web WakeUp?

1E Web Wake Up 1.5 provides enhanced security, computer search and name resolution:

- **Locked-down security** - The new version of 1E Web WakeUp now lets you register users who will be able to use the system to wake machines. Without the appropriate authorization users will not be able to search for or wake systems.
- **Enhanced computer search** - You can now search for computers using *domain/username* combinations. This increases the compatibility between 1E Web WakeUp and enterprise networks.
- **Increased accuracy** - 1E Web WakeUp is now able to resolve local computer names without relying on DNS. It does this via an ActiveX control added to the client browser on first access.
- **Extended API** - Users can now wake up to 10,000 machines in a single call, plus all the new security and search capabilities have been added to the 1E Web WakeUp API.
- **Greater scalability** - 1E Web WakeUp integrates with the Agility Framework database to store and locate registered machines

1.3 Conventions used in this guide

This section shows how to interpret the different styles used in this document to denote various types of information.

Cross references

Cross references are shown in italics. Cross references may be to diagrams or tables in the current document or to other documents.

Notes

Notes are shown in white on an orange background. For example, the following note provides some useful information:

NOTE: always pay attention to notes.

Code fragments and command-lines

This manual uses *Lucida Console* typeface plus a shaded background to denote code fragments and command-lines. For example, the following shows a 1E Web WakeUp installer command-line:

```
C:> msixexec /i webwakeUp.msi
```

1.4 Who is this guide for?

This guide is aimed at administrators and those who intend to access 1E Web WakeUp functionality programmatically, using the 1E Web WakeUp API, to integrate with 3rd party applications.

Section 2 Requirements

The requirements for installing and using 1E Web WakeUp are listed below.

2.1 1E Web WakeUp client requirements

1E Web WakeUp can be viewed from a machine with the following OS installed:

- Windows Vista (32- and 64-bit editions)
- Windows XP
- Windows 2000

The web application must be viewed using the following browsers:

- Internet Explorer 6.0 and above (32-bit version is required if the machine lookup ActiveX control is to be used; otherwise reverse DNS lookup must be used where 64-bit versions are deployed)

2.2 Server requirements

The 1E Web WakeUp server can be installed on a machine with the following OS installed:

- Windows Server 2008
- Windows Server 2003

The following items should be present prior to installing 1E Web WakeUp:

- IIS version 6.0 or later with .NET server extensions installed
- For IIS 7.0 we require IIS 6.0 metabase compatibility and Windows Authentication Role
- ASP.NET installed and enabled
- .NET Framework 3.5 SP1
- SMS 2003 SP2 or System Center Configuration Manager 2007 (optional)
- 1E Agility Framework 2.3.100.5 or later
- 1E WakeUp Product Pack 5.5.100.2 or later
- 1E WakeUp 5.5.1.166 or later using multi-agent mode

2.3 1E Web WakeUp service account

The following permissions are required for the 1E Web WakeUp service account.

Basic 1E Web WakeUp service account permissions

A domain user account for running the 1E Web WakeUp service is required during the installation of 1E Web WakeUp. Throughout this installation guide, this account will be referred to as *1EWWUSrv*. You should create this account before starting the installation process and grant permissions for the scenarios outlined in the following headings.

In all cases you will need to set the following privileges for the *1EWWUSrv* account by hand using Active Directory *Users and Computers* for the service account:

- Log on as service
- Access the computer from the network

After installation you will need to add permissions to the service account to enable access to the 1E Web WakeUp log directory, as described in *Section 4 - Installing 1E Web WakeUp* under heading *4.2 - Post Installation Configuration*

It is advisable that you also restrict the account in the following way:

- Deny logon locally

1E Web WakeUp service account DCOM permissions

You will also need to add the following DCOM access rights to *1EWWUSrv*:

- Run *DCOMCNFG*, navigate to *My Computer*, right-click and select *Properties*. In the *My Computer Properties* dialog select the *COM Security* tab.
- Under *Access Permissions*, click *Edit Limits*. Add the *1EWWUSrv* account and check *Remote Access*.
- Under *Launch and Activate Permissions*, click *Edit Limits*. Add the *1EWWUSrv* account and check *Remote Launch* and *Remote Activation*.

1E Web WakeUp service account WMI permissions

Permissions in the *N1E\WakeUp*, *SMSWAK* and *CIMV2* namespaces for the *1EWWUSrv* account must include the following security rights:

- Execute Methods
- Full Write
- Enable Account
- Remote Enable

The following steps show how to do this:

1. Run the WMI MMC plug-in (*wmimgmt.msc*), right-click on the root node and select *Properties*. In the *WMI Control (Local) Properties dialog* select the *Security* tab.
2. Navigate to the *N1E/WakeUp* node and click the *Security* button. In the *Security for ROOT\N1E\WakeUp* dialog add the *1EWWUSrv* account and check the permissions listed above.
3. Navigate to the *CIMV2* node and click the *Security* button. In the *Security for ROOT\CIMV2* dialog add the *1EWWUSrv* account and check the permissions listed above.
4. Navigate to the *SMSWak* node and click the *Security* button. In the *Security for ROOT\SMSWak* dialog add the *1EWWUSrv* account and check the permissions listed above.

2.4 1E Web WakeUp Installer account

The following permissions are required by the administrator installing 1E Web WakeUp.

Basic permissions

The administrator must have the following specific permissions assigned before starting the installation process.

- A domain account with administrator rights on the server being installed onto
- Sysadmin access to the Agility Framework Reporting SQL database
- When integrating with SMS/ConfigMgr the administrator needs to be an SMS/ConfigMgr Administrator. Their account will also need remote WMI permissions to the SMS/ConfigMgr namespace if the 1E Web WakeUp installation is remote from the SMS/ConfigMgr database

WMI permissions

Permissions in the *N1E\WakeUp* and *SMSWAK* namespaces for the administrator's account must include the following security rights:

- Execute Methods
- Full Write
- Enable Account
- Remote Enable

The following steps show how to do this:

1. Run the WMI MMC plug-in (*wmimgmt.msc*), right-click on the root node and select *Properties*. In the *WMI Control (Local) Properties dialog* select the *Security* tab.
2. Navigate to the *N1E\WakeUp* node and click the *Security* button. In the *Security for ROOT\N1E\WakeUp* dialog add the administrator's account and check the permissions listed above.
3. Navigate to the *SMSWak* node and click the *Security* button. In the *Security for ROOT\SMSWak* dialog add the administrator's account and check the permissions listed above.

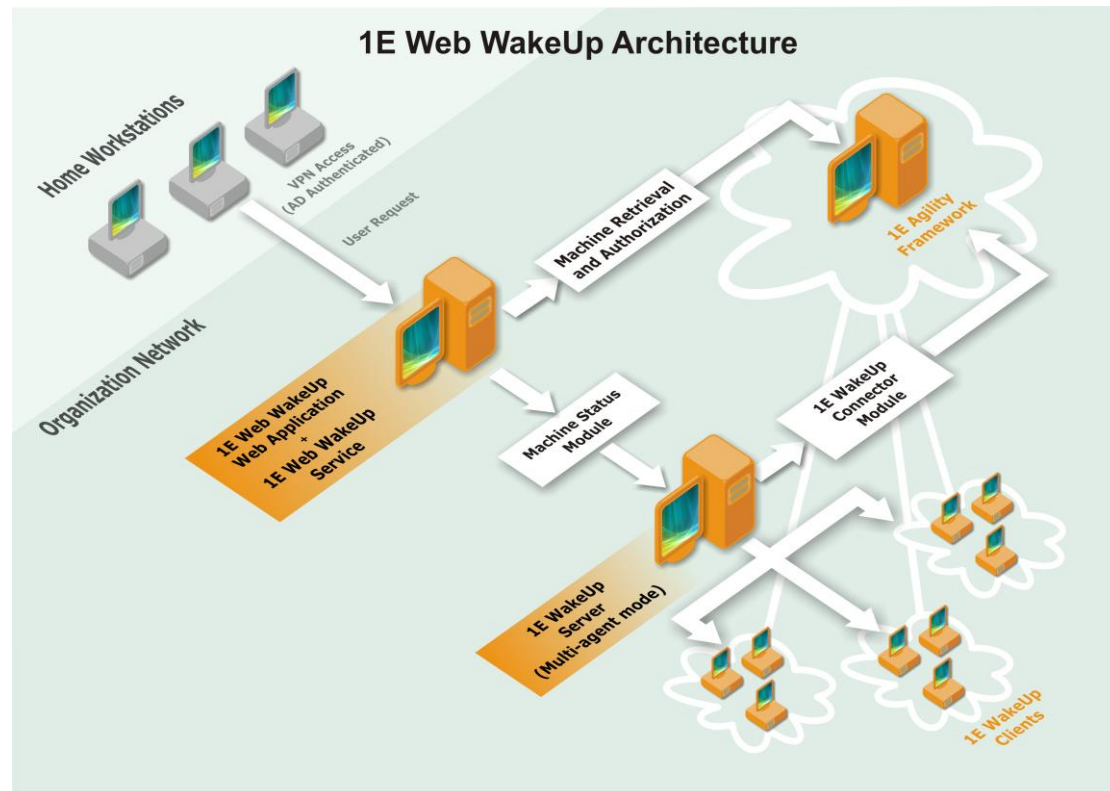
Section 3 1E Web WakeUp Overview

This section describes the 1E Web WakeUp Architecture and gives an overview on how 1E Web WakeUp performs its functions.

3.1 1E Web WakeUp Architecture

Figure 1 shows where the 1E Web WakeUp components reside and the way they interact with each other and the other elements of an organisation's network.

Figure 1 - The 1E Web WakeUp API Architecture



1E Web WakeUp consists of three parts:

- **The 1E Web WakeUp Web Application** – provides the interface to the application and communicates with the 1E Web WakeUp Service.
- **The 1E Web WakeUp Service** – retrieves machine details via the Machine Retrieval Module and communicates with the 1E WakeUp server via the Machine Status Module.
- **The 1E WakeUp Server** – retrieves information on machines via the 1E WakeUp Connector Module and distributes wakeups to target machines, usually via its 1E WakeUp agents. This includes the 1E WakeUp Provider that enables wakeups to be configured using WMI.

The 1E Web WakeUp Web Application and the 1E Web WakeUp Service should be located on the same server.

The wake up flow through the 1E Web WakeUp architecture

The following describes the flow through the 1E Web WakeUp architecture.

1. User, connected to the organisation's VPN, browses to the 1E Web WakeUp web site via http/https. The user's Windows credentials will have been authenticated via Active Directory.
2. User performs a function via the web interface: wake up registered machine; wake up named machine; search for machine to wake up or register machine.
3. The web site contacts 1E Web WakeUp service with the name of the machine to wake up, the search string or the name of the machine to register.

4. The Web Service calls out to a plug-in *Machine Retrieval* module to connect to a machine repository to collect matching machine details. When using strict authorization the 1E Web WakeUp authorization table is checked to see which machines the connected user is authorized to wake up.
5. If searching, a list of matching authorized machines is returned to the user.
6. The user selects the machine to wake up from the search results.
7. The 1E Web WakeUp Service signals the 1E WakeUp Server with the name of the machine to wake up.
8. The 1E Web WakeUp Service polls the status of the machine using a remote ping via the 1E WakeUp Server and returns the status to the web site.

Section 4 Installing 1E Web WakeUp

Prior to installation please check *Section 2 - Requirements* and make sure that all the criteria have been met.

1E Web WakeUp installs using MSI technology which requires *Windows Installer*; if this is not present it can be downloaded from <http://www.microsoft.com/downloads>. You can install interactively using the wizard interface provided, or you can install 1E WakeUp quietly without interaction by setting all the configuration options on the MSI command line.

Note: You must have Administrator rights or allow the files to be installed with elevated privileges on the system in order to install and configure 1E Web WakeUp.

4.1 Installing the 1E Web WakeUp Server using the installation wizard

To install you run the *WebWakeUp.msi* installer package. This can be done interactively following the steps in *Table 1*.

Table 1 - The 1E Web WakeUp installer interactive steps

Screen	Interaction
1. The Welcome Screen	No interaction required here, just click <i>Next</i> .
2. License Agreement	You should read the license information displayed here and then choose to continue with the installation by selecting the <i>I accept the license agreement</i> option and clicking on <i>Next</i> .
3. Destination Folder	Here you can change the 1E Web WakeUp installation directory or click <i>Next</i> to accept the default installation directory and settings. The default directory is: C:\Program Files\1E\webwakeUp\
4. Service Account	Sets the user name and password for the 1E Web WakeUp service account. The account name must be specified using the domain\user format. The account must have the permissions specified in <i>Section 2 - Requirements</i> under the heading <i>2.3 - 1E Web WakeUp service account</i> .
5. 1E WakeUp Provider	Here you set the location of the server where the 1E WakeUp Provider is installed. The 1E WakeUp Provider is the WMI interface to the 1E WakeUp server and is located on the same server. If this is the local server you can leave the default of <i>localhost</i> , otherwise you will need to provide the specific server name.
6. 1E Web WakeUp data	Here you set the location for the Agility Framework database and define whether 1E Web WakeUp should use strict authorization for performing wake ups. When strict authorization is enabled you will need to add entries to the Agility Framework database that list user names and machines that they can awaken before any wake ups can take place. See <i>Section 5 - Configuring 1E Web WakeUp Security</i> for more details.
7. Ready to Install the Application	The <i>Ready to Install the Application</i> screen is the final step before committing to an installation. Click <i>Next</i> to continue and install using the selected options. After the installation has taken place you will be shown the final screen of the wizard telling you that installation has been successful, click on <i>Finish</i> to end the wizard.

4.2 Post Installation Configuration

Once the 1E Web WakeUp installation process is completed, ensure that the 1E Web WakeUp service account has *read/write* access permissions on the log folder on the web service host machine.

C:\Documents and Settings\All Users\Application Data\1E\web.wakeup

You can leave any other default permissions, assigned to the service account when adding it to the permissions for this directory, as they are.

Section 5 Configuring 1E Web WakeUp Security

This section describes the new security model for 1E Web WakeUp. This allows you to lock down which users can wake which machines.

5.1 Enabling 1E Web WakeUp security

You can specify during installation to require authorization for users wanting to wake machines via the web interface.

Note: once you enable authorization, the users of 1E Web WakeUp will only be able to search for machines that they are associated with in the 1E Web WakeUp authorization database. If a user does not have an entry in the database they will not be able to search for, or wake, any machines.

5.2 Adding users to the 1E Web WakeUp security database

To allow users to wake up machines on the network you need to add their details to the 1E Web WakeUp authorized machines table. This is held in the agility framework database and is called *tbWWU_Authorised_UserMachines*. The columns are described in the following table:

Column	Description
UserTicket	The identifier for the user as returned by the authentication mechanism being used. So for Windows authentication this will be in the format <i>Domain\User</i> .
NetbiosName	The NetBIOS name for the machine that the specified user can wake up. Here you can use the * character as a specific shorthand for a "power user" type entry that specifies the user can wake any machine
DomainName	The domain name for the machine.

The following example data shows a user named *Joe* on the *ACME* domain who is allowed to wake the machines *ACMEDEV00017* and *ACMEDEV00015*; and a power user named *Mary* who is allowed to wake all the machines on the network.

UserTicket	NetbiosName	Domain
ACME\Joe	ACMEDEV00017	ACME
ACME\Joe	ACMEDEV00015	ACME
ACME\Mary	*	

The 1E Web WakeUp additions to the Agility Framework database also include a batch job that will populate the table from a .csv input file. More details on this method are provided in *Appendix C - Automatically setting the authorization table*.

5.3 Securing the 1E Web WakeUp Authorization web service

When using authorization you are also advised to lock down the 1E Web WakeUp Authorization Service in IIS. This web service provides the *AddAuthorisedMachine*, *DeleteAuthorisedMachine*, *GetAuthorisedMachines* and *UpdateAuthorisedMachine* methods that enable programmatic modification of the *tbWWU_Authorised_UserMachines* table. These methods are described in *Appendix A - The 1E Web WakeUp API*.

Locking down security

This is done by setting the NTFS file permissions on the file *AuthorisationServices.asmx* to restrict access so that only 1E Web WakeUp administrators can browse to the service. If access is not restricted then any user with web access and the appropriate knowledge can use the 1E Web WakeUp Authorization service to grant themselves or others permissions to wake any machine. In a default installation this file is located in *C:\Program Files\1E\WebWakeUp\Website\WebServices*.

Section 6 Using 1E Web WakeUp

This section describes the user interface to 1E Web WakeUp.

6.1 Logging on to 1E Web WakeUp

To access this page you browse to the web service using Internet Explorer using the following URL: `http://<servername>/WebWakeUp/Default.aspx`, where `<servername>` is the server where the 1E Web WakeUp API has been installed.

6.2 Installing the Web WakeUp ActiveX control

The first time you browse to the web service you will be asked if you want to install the Web WakeUp software published by 1E, as shown in *Figure 2*. The Active X control that gets installed when you click *Install* enables 1E Web WakeUp to confirm the name of the local computer, specifically for the purpose of registering the machine as the default wake up.

Figure 2 - The component installation dialog.



Note: If you choose not to install the ActiveX control 1E Web WakeUp will instead attempt to perform a reverse DNS lookup in order to derive the local computer name. The accuracy of this operation is dependent on DNS being correctly configured and up to date and may affect 1E Web WakeUp's ability to wake the computer at a later date.

6.3 The 1E Web WakeUp home page

Figure 3 shows the default page for the *ACMEServer/WebWakeUp* site with a user *coverlin* logged in to the *ACME* domain. This page shows that the user does not currently have a default wake up machine.

Figure 3 - The 1E Web WakeUp default page

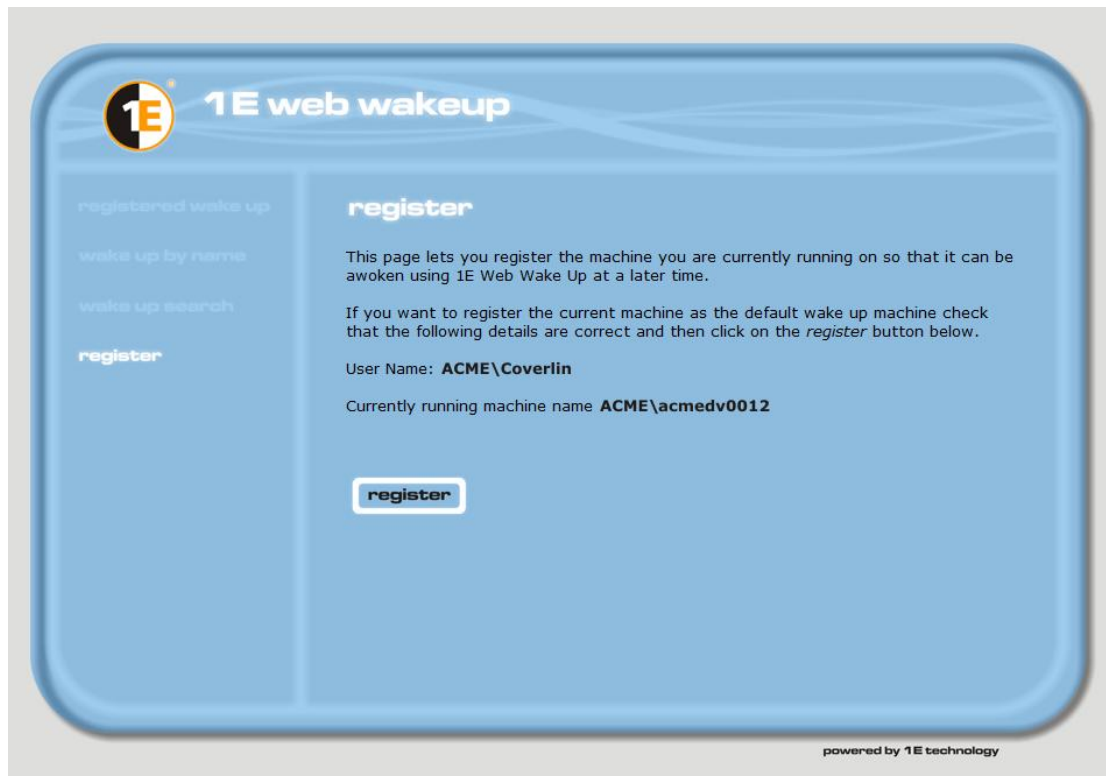


This will be the case when the user browses to the 1E Web WakeUp site the first time. In this case the user has three choices: they can register the machine they are currently running the browser on as the default wake up machine; they can enter an exact name for a machine to wake up or they can search for the machine name to wake up.

6.4 Registering a machine for Wake Up

One of the fundamental ideas behind 1E Web WakeUp is that the user does not need to remember the exact name of the machine they want to wake up remotely. All they need to do is go to the machine when it is up and running and register it via the 1E Web WakeUp interface. Then when the machine is turned off and they want to access it remotely, all they need to do is to browse to the 1E Web WakeUp interface from another machine and click on the *wake up* button, as described further on.

The register link on the left-hand side of the interface lets the user register the machine that the browser is currently running on for a future wake up. Clicking this link displays the *register* page, as shown in Figure 4, providing details of the logged on user and the machine name. To register the machine they click the *register* button.

Figure 4 - Registering the machine the user is currently running on

Once the machine is registered 1E Web WakeUp will confirm the result to the user, as shown in *Figure 5*.

Figure 5 - A successful registration

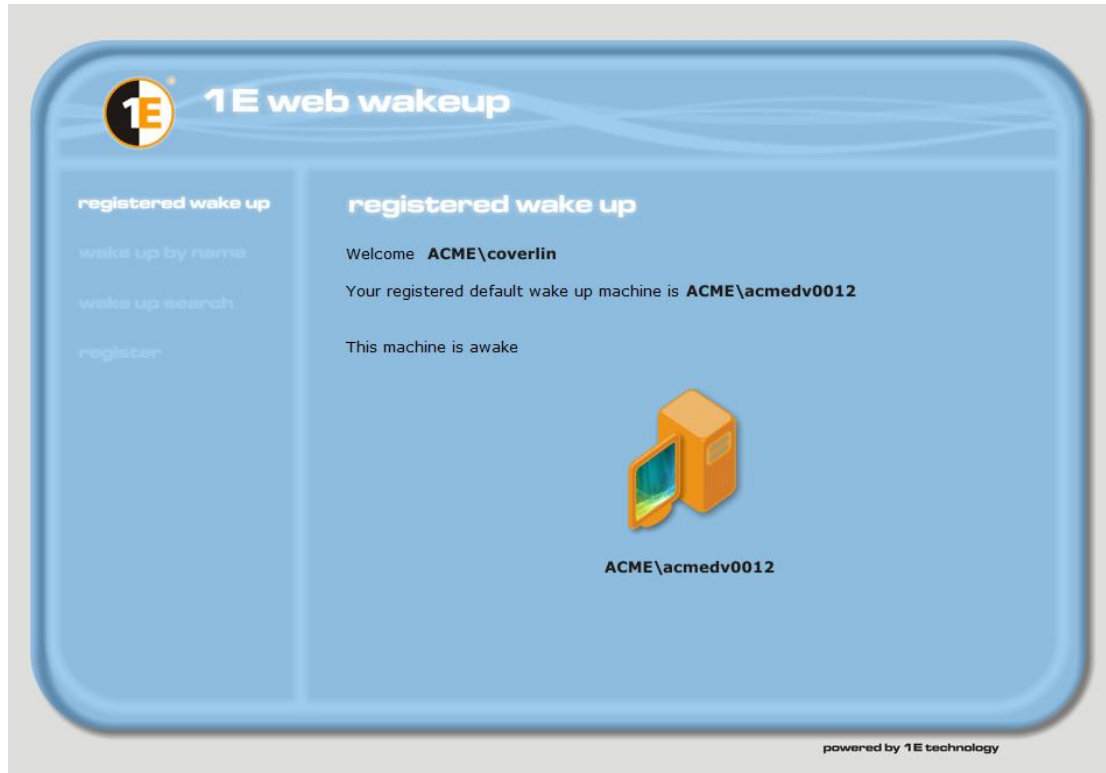
6.5 Waking up a registered machine

Once a machine has been registered for wake up, the next time the user visits the 1E Web WakeUp site the current status of the machine will be determined and then fed back to the user.

The registered machine is already awake

When the registered machine is already awake the user will see a page similar to the one shown in *Figure 6*, telling them their login details, the name of the registered machine and notification that the machine is already awake. The machine status icon at the bottom of the screen displays an orange *awake* status.

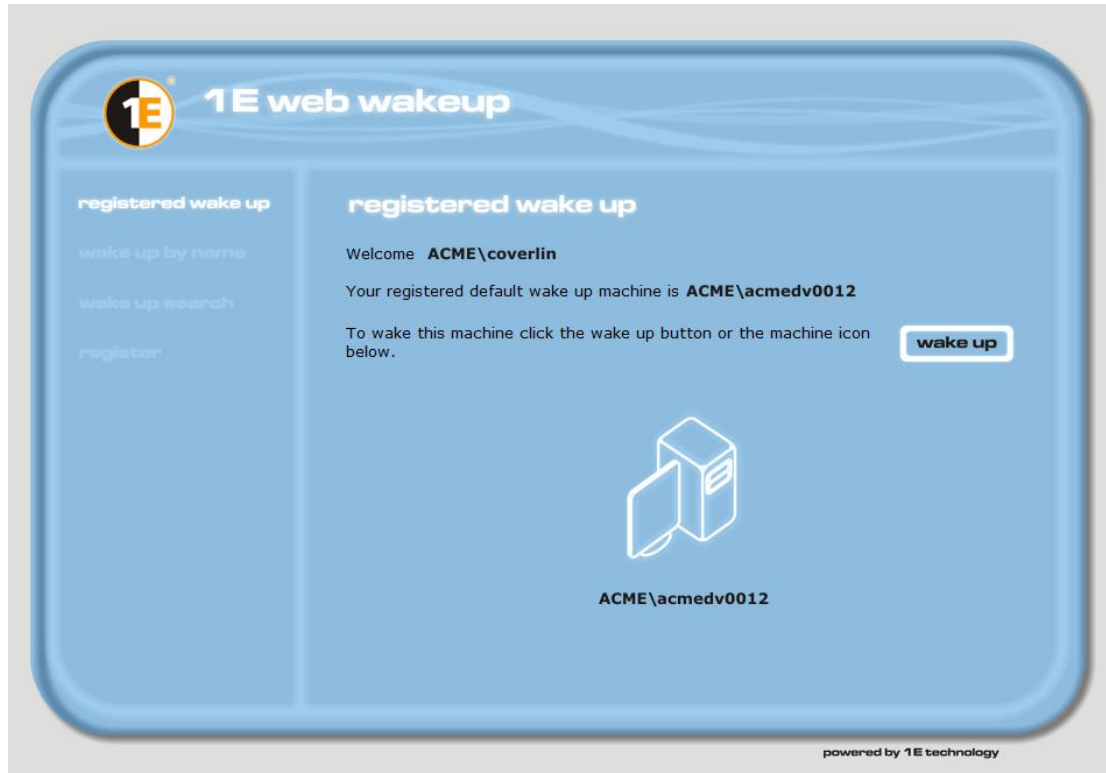
Figure 6 - A registered machine already awake



The registered machine is not awake

When the registered machine is not awake the user is notified of its status and is given the opportunity to wake the machine up. *Figure 7* shows the previously registered machine *acmedv0012* is not awake. To wake the machine up the user can either click on the *wake up* button on the right or the large machine status icon at the bottom of the console.

Figure 7 - The registered machine is not yet awake



Clicking on the *wake up* button initiates the attempt to wake the machine up. The wake up signal is sent to the machine and 1E Web WakeUp periodically checks the machine to see if it is now awake. The attempt to check to see if the machine is awake is done for a fixed period of time and the countdown for the timeout period is denoted by the in progress machine status icon. When all the bars around the edge of the progress icon have disappeared the period will be finished.

Figure 8 shows the attempt to wake up the machine *acmedv0012* about a quarter of the way through the timeout period. The machine might wake up at any point during the period depending on how fast the machine wakes.

Figure 8 - Attempting to wake the registered machine

If at any point the 1E Web WakeUp determines that the machine is awake, its status is returned to the user. *Figure 9* shows that the attempt to wake *acmedv0012* has been successful.

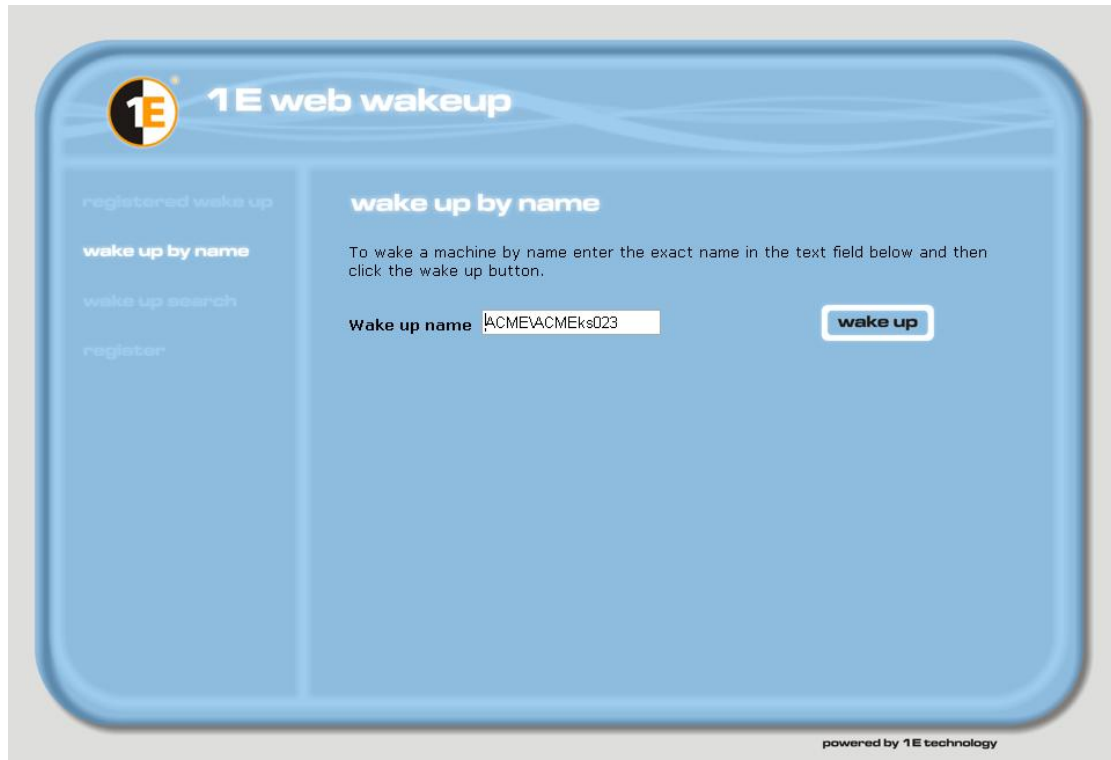
Figure 9 - Successfully waking the registered machine

6.6 Waking a machine by name

If the user wants to wake up a machine that is different from their registered default, or they do not have a registered default, as long as they know the exact name for the machine they can still wake it up. This is done from the *wake up by name* page, which they can get to by clicking on the wake up by name link on the left hand side of the console. This method also lets the user specify the domain name for the machine.

Figure 10 shows the wake up by name page with the name `ACME\ACMEks023` entered into the *wake up name* text field.

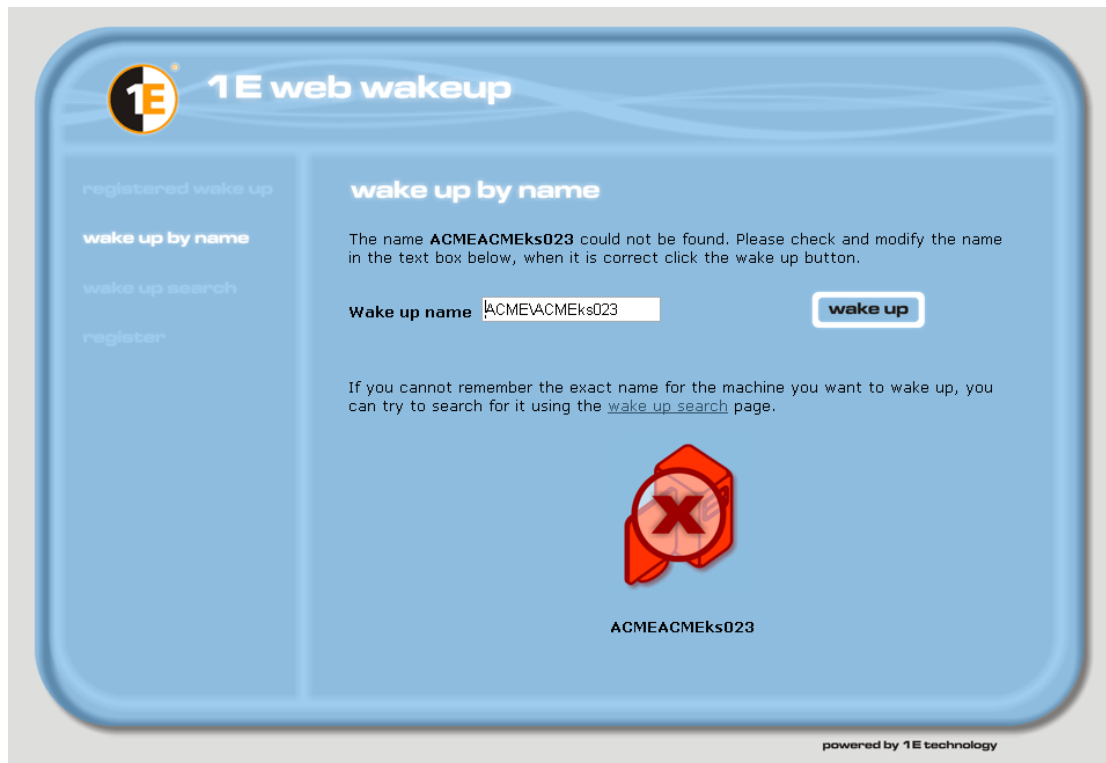
Figure 10 - Waking a machine using its exact name



Once they have entered the name for the machine the user then clicks on the *wake up* button to initiate the wake up.

Incorrect wake up names

When the wake up name is wrong 1E Web WakeUp will not be able to retrieve any information on the machine to be awoken. When this happens the only choice is for the user to correct the name that has been input. So 1E Web WakeUp notifies the user with an appropriate notification and gives them the opportunity to make a change to the entered name. In this case the machine status icon at the bottom of the console will change to one that represents the fact that the name is incorrect. Figure 11 shows that the `ACMEks023` machine name entered previously could not be found by 1E Web WakeUp.

Figure 11 – When the machine name is not correct

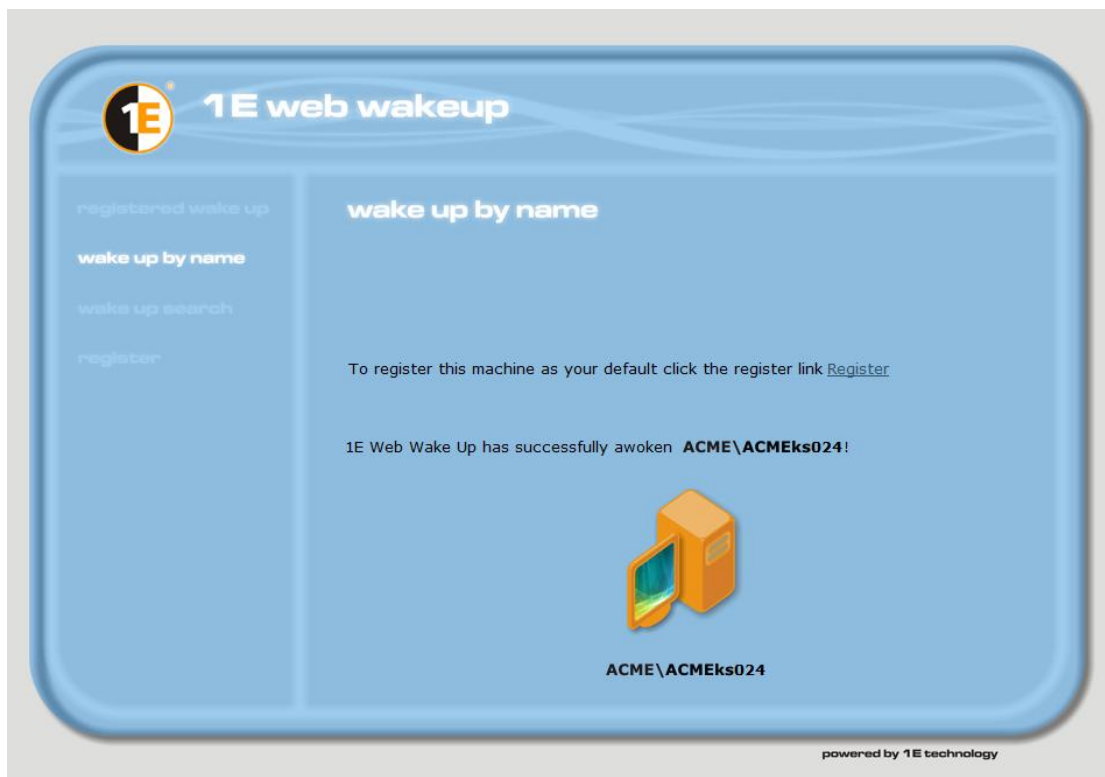
If the user can see how the name is incorrect they then correct the name and click on the *wake up* button again. *Figure 12* shows that the user correcting the name to *ACMEks024*. They then click on the *wake up* button to try again.

Figure 12 - Correcting the name of the machine

This time the machine name is found by 1E Web WakeUp and the attempt to wake the machine starts, as shown in *Figure 13*.

Figure 13 - Attempting to wake the altered machine name

Finally the altered machine wakes up successfully, as shown in *Figure 14*.

Figure 14 - The machine wakes up successfully

6.7 Waking a machine using search

If the user wants to wake a machine that is different from their registered machine, or there is no registered machine and the user is unsure of the name, they can instead use a search string to return names that are similar to any parts of it that they do recall. This is done on the *wake up search* page, which can be displayed by clicking on the *wake up search* link on the left hand side of the console. Here the user can enter the search text in the text field and then click the *search* button to initiate the search.

Domain and machine name search

The search text does not allow the use of wildcards but it does enable the entry of a single search string that will match against the domain and machine name fields separately. The rules that govern the search are as follows:

- Searches are not case sensitive
- Backslash "\" separates the domain name search string from the machine name search string. If the \ character is omitted the search text will be performed on the machine name only. The format of the search string is *DomainSearch\MachineSearch*.
- Empty search strings return all entries

The search text matches against presence of the search characters in the relevant field. So the search text *XP* will match machines whose netBIOS names contain the letters *XP* in that order such as *XP01034*, *Experience1*, *vaxpac89* etc...

Searching for domains is done by adding the domain search at the start with a backslash. So the search text *M\1* will return all the machines in domains that contain the letter *M* whose machine names contain the number *1*.

If you want to return all the items in a particular field you can leave the search string empty. So the search text *M* will return all machines in domains that contain the letter *M*. Similarly leaving the search field entirely blank will retrieve all machines in all domains.

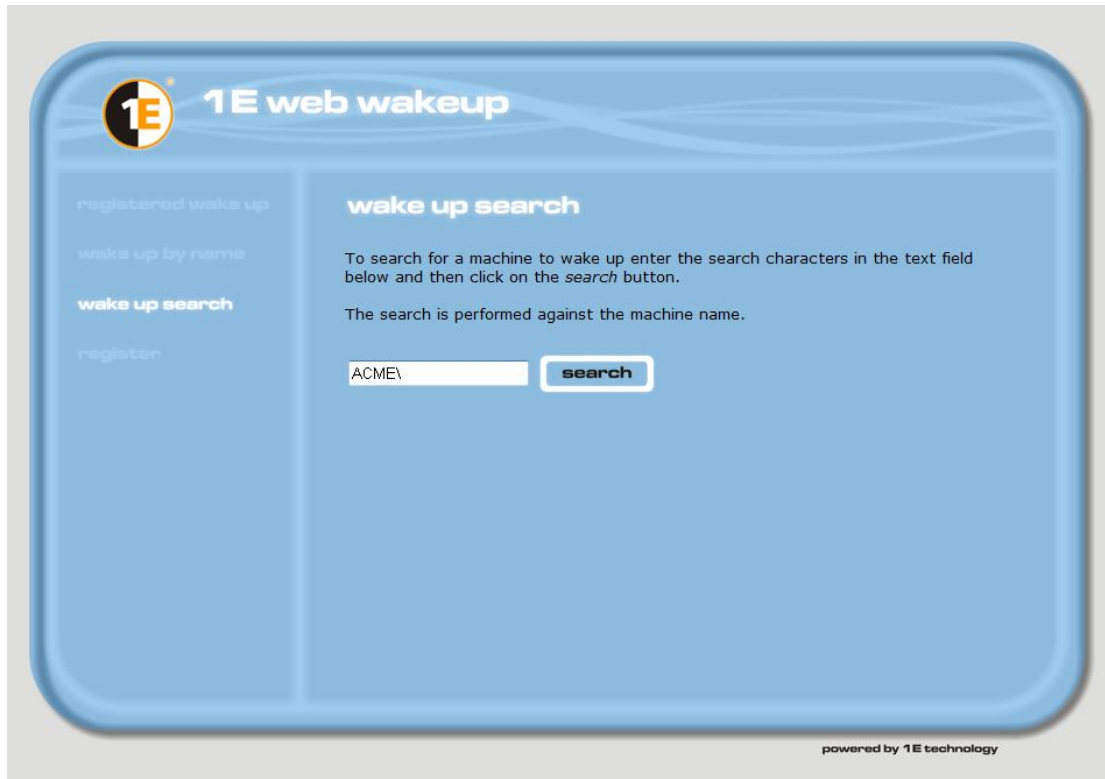
Search result limits

The number of machines returned by a search is limited by default to 30. The results are not paged so if the required machine is not initially shown the search will need to be refined. A 1E Web WakeUp administrator can raise the limit on a site-wide basis up to a maximum of 500, as described in *Appendix B - Configuration Options*, under the heading *Setting the maximum number of machines returned*, but this may impact on page loading times if a general search is performed on a large network.

An example search

Figure 15 shows the user *coverlin* entering a search string that will retrieve all machines on the *ACME* domain.

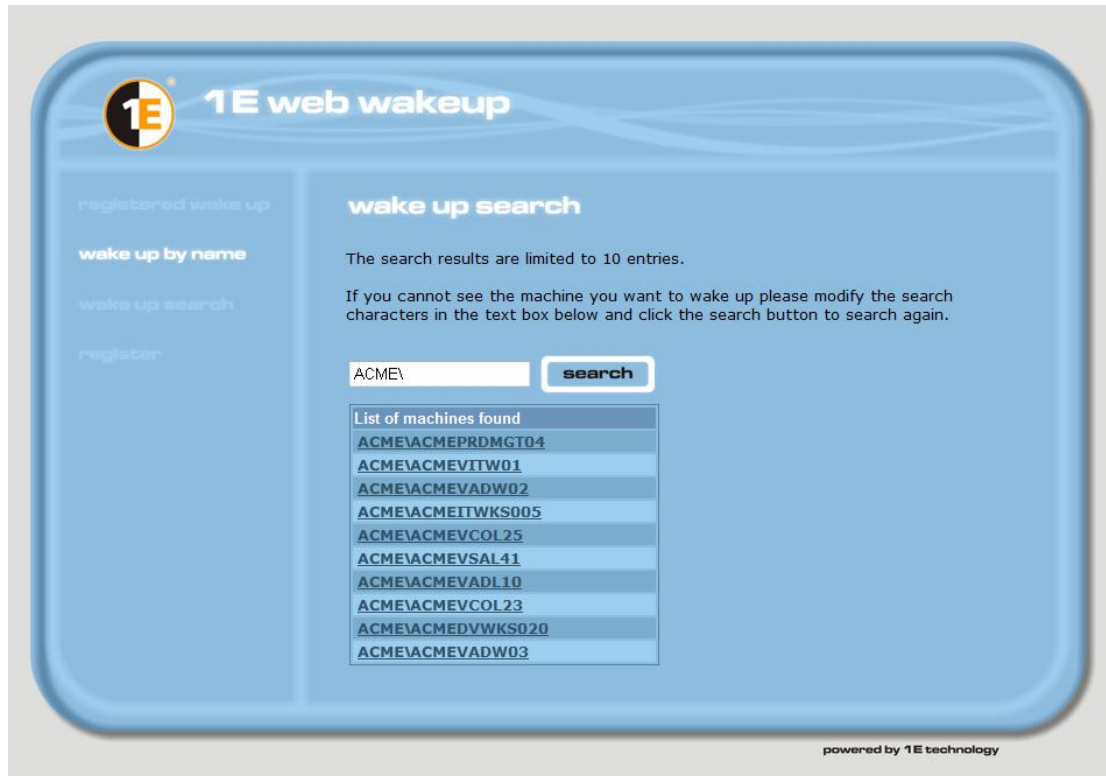
Figure 15 - Searching for a machine name



When the search button is clicked 1E Web WakeUp will return machines whose names match the search string; the search is not case sensitive. The number of machines returned by the search is configurable.

Figure 16 shows the search results for the *ACME* string entered earlier.

Figure 16 - The search results



To wake a machine from the search results the user just clicks on the name of the machine. *Figure 17* shows what happens after the user clicks on the *ACMEVADL10* machine name. The attempt to wake the machine has started and continues to count down in the same way as for the registered wake up and wake up by name operations.

Figure 17 - Attempting to wake a machine from the search results



A wake up timeout

In our example the timer ticks down all the way to zero and the user's attempt to wake the *ACMEVADL10* machine times out. When this is the case the user is notified and the machine status icon changes accordingly, as shown in *Figure 18*.






A timeout does not necessarily mean that there has been an error as the machine may just have been slow in starting up. In this case it is recommended that the user tries the machine again after a reasonable time. If the machine still does not respond then they should contact their network administrator to see if there is a problem with the machine or network.

Figure 18 - When 1E Web WakeUp is unable to determine the state of the machine



6.8 The machine status icon

The status of registered machines and the attempts to wake machines by name or by search is signalled by the machine status icon that appears at the bottom of the console. The following table shows the possible states and the icon used to represent the state:

State	Icon
<p>The machine is awake.</p> <p>The user does not need to take any action.</p>	
<p>The machine is not awake.</p> <p>When the machine is in this state the user can attempt to wake the machine up by clicking on this icon.</p>	
<p>The attempt to wake the machine timed out. This only signifies that 1E Web WakeUp was unable to determine the wake state for the machine.</p> <p>In this case the user should wait for a suitable period before attempting to communicate with the machine.</p>	
<p>The machine name entered is incorrect.</p> <p>In this case the user may want to modify the name and try again.</p>	
<p>There was a problem when attempting to communicate with the machine. This may be due to a network problem or a problem with the 1E WakeUp server.</p> <p>In this case the user will need to contact the network administrator to remedy the problem.</p>	

The machine wake up progress icon

1E Web WakeUp displays a progress icon, as shown in *Figure 19*, during the attempt to wake up a machine. This icon shows that an action is taking place and counts down to the end of the period during which 1E Web WakeUp attempts to determine whether the machine is awake or not. As the time is counting down the bars around the edge of the icon disappear until all the bars disappear and the attempt times out.

Figure 19 - The machine wake up progress icon

Section 7 Troubleshooting

When troubleshooting problems with the 1E Web WakeUp API you should first check that the system meets the requirements set out in the *Section 2 - Requirements*. If the requirements are met you should then follow through the process for creating a problem report to send to the 1E technical support team shown below.

7.1 Checking the problem

Before contacting 1E technical support with a problem you should check the following:

- Make sure that the minimum requirements have been met.
- Make sure that the problem is reproducible after a machine reboot (if possible) or after stopping and restarting the service.
- Read the *readme.htm* file included with the 1E Web WakeUp installation. This includes late breaking news, and details on known issues.

7.2 Environmental issues

The following are known possible environmental issues that may cause problems when using 1E Web WakeUp:

- If you are encountering problems with waking up registered machines when users browse to the site using 64-bit versions of Windows Explorer, it could be due to the DNS entry in the domain being incorrect, leading to the wrong machine becoming registered for the given name.

1E Web WakeUp provides an ActiveX control, available when using the 32-bit version of Internet Explorer, which will correctly determine the server name this is installed the first time an end user browses to the web site.

If you are unable to switch the users over to the 32-bit version of Internet Explorer and on to the ActiveX control, you will need to check your DNS settings if this issue is occurring regularly.

7.3 Contacting 1E support

If you encounter problems with using 1E Web WakeUp and you are registered with 1E Technical Support, the best action is to contact the 1E technical support team who will help 1E determine the solution to your problem quickly. To help speed the process you should create a technical report of the problem.

Creating a technical report

1E Web WakeUp log files are vital for technical support. To help determine the cause of problems please ensure that log files are submitted. 1E Web WakeUp log files are saved to the common application data folder. This is typically:

```
C:\Documents and Settings\All Users\Application Data\1E\web.wakeup
```

Please ensure that this log file covers the period during which the issue occurred. If not, please delete the log file and re-run the steps to replicate the issue. This will create a new log file relevant to the problem and enable our technical representatives to help more efficiently.

The technical report should contain the following information:

- The 1E WebWakeUp service log file *WakeUp.Web.Services.log*
- Version numbers for 1E WebWakeUp
- The OS, version number and patch level the machine is running.
- The version number and patch level of SMS/ConfigMgr (if used).
- The version number of 1E Agility Framework
- Please also list the steps taken to cause the error, so that we can reproduce it.

You should then email the details of the problem encountered along with the above information to support@1e.com. A technical consultant will then contact you to help find a suitable solution.

Section 8 Further Information

For more detailed questions about specific situations that may be relevant to your network you are always welcome to contact 1E directly. Our contact details are provided below.

8.1 Contact details

1E provide a number of SMS enhancement tools as well as consultancy services. This section provides information on how to get information about 1E and contact details for the various departments within 1E.

Website

The essential resource for information about 1E and its products is the website.

www.1e.com

Telephone and fax

The following UK numbers are available for telephone and fax enquiries:

Tel: +44 (0)20 8326 3880

Fax: +44 (0)20 8840 9578

The following number is available in the United States of America:

Tel: 1-800 516 6938

Post

The postal address for 1E is:

Head Office:
1E Ltd,
97-107 Uxbridge Rd,
London W5 5TL, UK

Sales

To contact the sales department at 1E you can use the following email address:

sales@1e.com

Technical support

To contact the technical support department at 1E you can use the following address:

support@1e.com

Appendix A The 1E Web WakeUp API

1E Web WakeUp provides an interface that contains some functions that allow you to find and wake a named machine and check to see its current status. This appendix describes each of the functions and how to create service and web references.

The information provided here is for advanced users with a pre-requisite knowledge of web service development.

A.1 Using the 1E Web WakeUp API

There are two ways of using the 1E Web WakeUp API: one uses Microsoft Visual Studio to create a service reference to the web service URL; the other uses a web reference.

Creating a service reference

In Microsoft Visual Studio 2008 you can add a *Service Reference* to the web service URL <http://host/WebWakeUp/WebServices/LocalServices.asmx>.

This generates a proxy class, typically with the name *LocalServicesSoapClient*. In code the proxy class can then be used to access the 1E Web WakeUp API methods. For example the following code fragment shows the *FindMachines()* method being invoked:

```
var client = new lssc.LocalServicesSoapClient();
try
{
    var resultArr = client.FindMachines("machineName");
    //do what you want to do with the results array.
}
catch(Exception ex)
{
    //handle the error case
}
finally
{
    client.Close(); //MUST CALL THIS
}
```

Note: the code must call the *client.Close()* method after use.

Creating a web reference

In Microsoft Visual Studio 2005 and Microsoft Visual Studio 2008, there is an advance option of the service reference called a *web reference*. For 1E Web WakeUp you can add a *Web Reference* to the web service URL <http://host/WebServices/LocalServices.asmx>.

This generates a proxy class to the web service, typically with name *LocalServicesWse*. In code the class can then be used to access the 1E Web WakeUp API methods. For example the following code fragment shows a reference to *LocalServices.asmx* that has been created with the name *lssc* and the *FindMachines()* method being invoked on the new soap client that is created via the reference:

```
var client = new lssc.LocalServicesSoapClient();
String[] result = client.FindMachines("MachineName");

finally
{
    client.Close(); //MUST CALL THIS
}
```

As 1E Web WakeUp uses windows authorization, the basic HTTP binding in the application's *app.config* file will also need the following security settings added:

```
...
<security mode="TransportCredentialOnly">
    <transport clientCredentialType="Ntlm"/>
</security>
...
```

A.2 The 1E Web WakeUp API Method Reference

This heading details the methods that are available in the 1E Web WakeUp API. These methods become available for use once an instance of the 1E Web WakeUp API class has been created.

Below is a specific reference for each method available in the 1E Web WakeUp API. Each reference contains a prototype with inputs and returns. The argument names are shown in italics; the types in normal weight and the method name is shown in bold.

Note: all arguments are mandatory.

AddAuthorisedMachine

Description	Add a machine to a user's authorized list of machines
Prototype	bool AddAuthorisedMachine (string <i>UserName</i> , string <i>MachineName</i>)
Inputs	<p><i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i>.</p> <p><i>MachineName</i> The domain name for the machine to be added. This should be in the format <i>DomainName\MachineName</i>.</p>
Returns	Boolean denoting success or failure.
Remarks	
Example (C#)	<pre>var client = new lssc.LocalServicesSoapClient(); try { var result = client.AddAuthorisedMachine("ACME\Joe", "ACME\ACMEDvws0012"); if (result) Console.WriteLine("added ACMEDvws0012 for Joe"); else Console.WriteLine("could not add ACMEDvws0012"); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS }</pre>

ClearRegisteredMachine

Description	Clear the default machine registered to a user.
Prototype	bool ClearRegisteredMachine (string <i>UserName</i>)
Inputs	<i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i> .
Returns	Boolean denoting success or failure.
Remarks	
Example (C#)	<pre>var client = new lssc.LocalServicesSoapClient(); try { var result = client.ClearRegisteredMachine("ACME\Joe"); if (result) Console.WriteLine("registered machine cleared"); else Console.WriteLine("could not clear machine"); } catch(Exception ex) { //handle the error case }</pre>

```

}
finally
{
    client.Close(); //MUST CALL THIS
}

```

DeleteAuthorisedMachine

Description	Deletes a machine from a user's authorized list of machines using its name.
Prototype	bool DeleteAuthorisedMachine (string <i>UserName</i> , string <i>MachineName</i>)
Inputs	<p><i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i>.</p> <p><i>MachineName</i> The name for the machine to be deleted. This should be in the format <i>DomainName\MachineName</i>.</p>
Returns	Boolean denoting success or failure.
Remarks	
Example (C#)	<pre> var client = new lssc.LocalServicesSoapClient(); try { var result = client.DeleteAuthorisedMachine("ACME\Joe", "ACME\ACMEDvwnks0012"); if (result) Console.WriteLine("removed ACMEdvwnks0012 from Joe"); else Console.WriteLine("could not remove ACMEdvwnks0012"); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS } </pre>

FindMachineExact

Description	Find a machine using its exact netbios name.
Prototype	bool FindMachineExact (string <i>MachineName</i>)
Inputs	<p><i>MachineName</i> The netbios name for the machine to be found. Anything but a complete match will result in failure. Can also be in the format <i>DomainName\MachineName</i>.</p>
Returns	Boolean denoting success or failure.
Remarks	This method is used to find when the exact netbios name for the target machine is known.
Example (C#)	<pre> var client = new lssc.LocalServicesSoapClient(); try { var result = client.FindMachineExact("ACME\ACMEDvwnks0012"); if (result) Console.WriteLine("found ACMEdvwnks0012"); else Console.WriteLine("cannot find ACMEdvwnks0012"); } catch(Exception ex) { //handle the error case } </pre>

	<pre>finally { client.Close(); //MUST CALL THIS }</pre>
--	---

FindMachines

Description	Find a number of machines matching a search string.
Prototype	string[] FindMachines (string <i>SearchText</i>)
Inputs	<i>SearchText</i> The search text matched against the netbios name for the machines to be found. Can also be in the form <i>DomainName\SearchText</i>
Returns	Array of strings containing the machines that match the search text. The number of items in this array will be limited by the maximum number of returned items allowed. This can be found using the
Remarks	This method is used to find when the exact netbios name for the target machine is not known.
Example (C#)	<pre>var client = new lssc.LocalServicesSoapClient(); try { var resultArr = client.FindMachines("ACME\ACME"); //process the results array } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS }</pre>

FindRegisteredMachine

Description	Find the default machine registered to a user.
Prototype	string FindRegisteredMachine (string <i>UserName</i>)
Inputs	<i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i> .
Returns	String containing the name of the machine registered to the user.
Remarks	
Example (C#)	<pre>var client = new lssc.LocalServicesSoapClient(); try { var result = client.FindRegisteredMachine("ACME\Joe"); if (result == "") Console.write("no registered machine for Joe"); else Console.write(result); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS }</pre>

GetAuthorisedMachines

Description	Get an array of machine names on a user's authorized list of machines.
Prototype	String[] GetAuthorisedMachines (string <i>UserName</i>)
Inputs	<i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i> .
Returns	A string array containing all the machines the user is authorized to wake up.
Remarks	
Example (C#)	<pre> var client = new lssc.LocalServicesSoapClient(); try { var result = client.GetAuthorisedMachines("ACME\Joe"); //process results array } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS } </pre>

GetMaxItemCount

Description	Get the maximum number of items returned by 1E Web WakeUp
Prototype	int GetMaxItemCount ()
Inputs	
Returns	Returns an integer that gives the maximum number of items that will be returned in the array returned by the <i>FindMachines</i> method.
Remarks	
Example (C#)	<pre> var client = new lssc.LocalServicesSoapClient(); try { var i = client.GetMaxItemCount(); console.WriteLine("The max items value: {0}", i); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS } </pre>

Ping

Description	Check to see if a named machine is awake using a ping.
Prototype	bool Ping (string <i>NetBiosName</i>)
Inputs	<i>NetBiosName</i> The netbios name for the machine to ping.
Returns	Returns a boolean denoting success or failure of the ping.
Remarks	<p>This method is provided as a useful utility method for occasional use. It is not recommended to use the <i>Ping</i> method on large numbers of machines.</p> <p>The <i>Ping</i> method cannot determine if a machine is awake if any intermediate firewalls block ICMP. However the ping request is sent by the 1E WakeUp server so the local machine may still have ICMP firewalled and the 1E WakeUp server inside the firewall will still be able to function.</p>
Example (C#)	<pre>var client = new Issc.LocalServicesSoapClient(); try { var result = client.Ping("ACMEdvwks0012"); if (result) Console.WriteLine("response for ACMEdvwks0012"); else Console.WriteLine("no response for ACMEdvwks0012"); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS }</pre>

RegisterMachine

Description	Add the default registered machine for a user using its name.
Prototype	bool RegisterMachine (string <i>UserName</i> , string <i>MachineName</i>)
Inputs	<i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i> . <i>MachineName</i> The netbios name for the machine to be set.
Returns	Boolean denoting success or failure.
Remarks	
Example (C#)	<pre>var client = new Issc.LocalServicesSoapClient(); try { var result = client.RegisterMachine("ACME\Joe", "ACME\ACMEdvwks0012"); if (result) Console.WriteLine("ACMEdvwks0012 default for Joe"); else Console.WriteLine("could not set ACMEdvwks0012"); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS }</pre>

UpdateAuthorisedMachine

Description	Update the name of a currently authorized machine for a user.
Prototype	bool UpdateAuthorisedMachine (string <i>UserName</i> , string <i>MachineName</i> , string <i>OldMachineName</i>)
Inputs	<p><i>UserName</i> The user name for an authorized user. This should be in the format <i>DomainName\UserName</i>.</p> <p><i>MachineName</i> The new name for the machine being updated. This should be in the format <i>DomainName\MachineName</i>.</p> <p><i>OldMachineName</i> The old name for the machine being updated. This should be in the format <i>DomainName\MachineName</i>.</p>
Returns	Boolean denoting success or failure.
Remarks	
Example (C#)	<pre>var client = new Issc.LocalServicesSoapClient(); try { var result = client.UpdateAuthorisedMachine("ACME\Joe", "ACME\ACMEDvws0012", "ACME\ACMEDvws0017"); if (result) Console.WriteLine("updated ACMEdvws0012"); else Console.WriteLine("could not update ACMEdvws0012"); } catch(Exception ex) { //handle the error case } finally { client.Close(); //MUST CALL THIS }</pre>

WakeByName

Description	Attempt to wake a machine using its netbios name
Prototype	bool WakeByName (string <i>NetBiosName</i>)
Inputs	<i>NetBiosName</i> The netbios name for the machine to wake up.
Returns	Returns a boolean denoting the success or failure of the sending of the wake up request.
Remarks	Note: If you intend to make a high volume of calls to wake machines in a short space of time you should use the <i>WakeMachines</i> method, which supports waking a list of machines, and not <i>WakeByName</i>.
Example (C#)	<pre>var client = new Issc.LocalServicesSoapClient(); try { var result = client.WakeByName("ACMEDvws0012"); if (result) Console.WriteLine("sent wake up to ACMEdvws0012"); else Console.WriteLine("no wakeup sent to ACMEdvws0012"); } catch(Exception ex) { //handle the error case } finally { }</pre>

	<pre>client.close(); //MUST CALL THIS }</pre>
--	---

WakeMachines

Description	Attempt to wake a number of machines on a list
Prototype	bool WakeMachines (string <i>MachineNames</i>)
Inputs	<i>MachineNames</i> A list of machine names to wake up.
Returns	Returns a boolean denoting the success or failure of the sending of the wake up requests.
Remarks	<p>The <i>MachineNames</i> list is a comma separated list of machines. This must be in the form "<i>domainA\name1, domainB\name2, ...</i>"</p> <p>There is a limit on the number of machines that can be awoken in a single call governed by the maximum length of the input argument. 1E WakeUp itself has an absolute upper limit of 10,000 machines that can be awoken in a single operation, but given the default value for the <i>WakeMachines</i> input argument length of 65536 characters and an average <i>domain\machine</i> name length of around 26 characters the default limit on <i>WakeMachines</i> is around 2500 machines.</p> <p>To increase the default input argument size you can change some bindings in the WakeUp.Web.Services.exe.config file, as described in <i>Appendix B - Configuration Options</i> under the heading <i>Setting the size of the machine search buffers</i>.</p>
Example (C#)	<pre>var client = new lssc.LocalServicesSoapClient(); try { var result = client.wakeMachines("ACMEDvws0012, ACMEDvws0013, ACMEDvws0014"); if (result) Console.WriteLine("sent wake up to machines"); else Console.WriteLine("no wakeup sent to machines"); } catch(Exception ex) { //handle the error case } finally { client.close(); //MUST CALL THIS }</pre>

Appendix B Configuration Options

1E Web WakeUp supports a small number of administrator configurable options. Some of these are set in the *WakeUp.Web.Services.exe.config* file located in the 1E Web WakeUp installation directory. By default this is in *C:\Program Files\1E\WebWakeUp\Service*. Others reside in the *web.config* file which by default is located in *C:\Program Files\1E\WebWakeUp\Website*.

The configuration file uses XML format and strict observance of the existing syntax must be followed.

B.1 Implementing changes

After changing the configuration options you must perform an IIS reset before the changes take effect. Most of the configuration file provides internal settings that should not be changed from the default. The following headings show the ones that may be changed under specific circumstances to modify the behaviour of 1E Web WakeUp.

Intervening settings in the following example configuration file fragments have been omitted and are represented by the characters "...". You will need to include any omitted settings, unchanged where appropriate, in your config file.

B.2 WakeUp Web Services options

This section shows the administrator configurable options in the *WakeUp.Web.Services.exe.config* file. After changing these settings you will need to restart the 1E Web WakeUp Service for them to take effect.

Using SMS/ConfigMgr as the source machine database

Data on machines is only collected by 1E WakeUp when it is configured for multi-agent use. If you are using 1E WakeUp in single agent mode you will need to configure 1E Web WakeUp to use an alternative for the source machine database, required in order to be able to wake machines.

The only alternative currently supported is the use of SMS/ConfigMgr. The options you need to change in the *WakeUp.Web.Services.exe.config* file are described in the following table:

Key name	Description
CurrentAddInAssemblyName	This will need to be changed from the default of: <i>"N1E.WakeUp.Web.AddInModule.AFRDataSourceAddIn.dll"</i> Where the information is retrieved from the Agility Framework Reporting database, to the following: <i>"N1E.WakeUp.Web.AddInModule.SMSDataSourceAddIn.dll"</i> Where the information is retrieved from SMS/ConfigMgr
SMSProvider	Set this to the server name where the SMS/ConfigMgr provider is installed
SMSSiteCode	Set this to the SMS/ConfigMgr site code

The following configuration file fragment shows an example where the SMS/ConfigMgr provider name is *ACME01* and the SMS/ConfigMgr site code is *S01*.

```

...
<appSettings>
  ...
  <add key="CurrentAddInAssemblyName"
        value="N1E.wakeUp.Web.AddInModule.SMSDataSourceAddIn.dll"/>
  ...
  <add key="SMSProvider" value="ACME01"/>
  <add key="SMSSiteCode" value="S01"/>
  ...
</appSettings>
...

```

Setting the maximum number of machines returned

When searching for machines in the 1E Web WakeUp website a default maximum of 30 machines is returned. The results are not paged so the only way of allowing users to view more machines is by changing the number of machines returned by 1E Web WakeUp.

The options you need to change in the *WakeUp.Web.Services.exe.config* file are described in the following table:

Key name	Description
MaxMachines	The default for this setting is "30". The minimum is "10" and the maximum is "500".

The following configuration file fragment shows an example where the maximum returned machines is set to 100.

```
...
<appSettings>
  ...
  <add key="MaxMachines" value="100"/>
  ...
</appSettings>
...
```

Turning authorization on or off

1E Web WakeUp supports setting the use of authorization which can be selected at installation. To modify authorization after installation you will need to edit the configuration file. The following table shows the key that must be changed to turn authorization on or off.

Key name	Description
AuthorisedMachineModule	To turn authorization on you should set this to: "N1E.WakeUp.Web.AuthorisedMachineModule.dll" To turn authorization off you should set this to: ""

The following configuration file fragment shows an example where authorization is turned on.

```
...
<appSettings>
  ...
  <add key="AuthorisedMachineModule"
    value="N1E.wakeUp.Web.AuthorisedMachineModule.dll"/>
  ...
</appSettings>
...
```

The following configuration file fragment shows an example where authorization is turned off.

```
...
<appSettings>
  ...
  <add key="AuthorisedMachineModule" value=""/>
  ...
</appSettings>
...
```

Note: once you enable authorization, the users of 1E Web WakeUp will only be able to search for machines that they are associated with in the 1E Web WakeUp authorization database. If a user does not have an entry in the database they will not be able to search for, or wake, any machines.

Setting the size of the machine search buffers

The *WakeMachines* method takes an input argument that contains a comma separated list of machines. This limits the number of machines that can be awoken in a single call to the maximum length of the input argument string. There is an absolute upper limit of 10,000 machines that can be awoken by 1E WakeUp in a single operation but given the default value for the *WakeMachines* input argument string length of 65536 characters and an average *domain\machine* name length of around 26 characters the limit on *WakeMachines* is around 2500 machines.

To increase raise the limit you can make a change to the following bindings in the *WakeUp.Web.Services.exe.config* file.

Binding Name	Description
MachineSearchServiceBinding	This binding contains two properties: <i>maxBufferSize</i> and <i>maxReceivedMessageSize</i> which are both set by default to 65536. When changing values you need to set both these properties to the same value otherwise the service will raise an error on startup

The following configuration file fragment shows an example where the buffer size has been increased to 131072.

```

...
<bindings>
...
  <netNamedPipeBinding>
...
    <binding name="MachineSearchServiceBinding"
      maxBufferSize="131072"
      maxReceivedMessageSize="131072" />
...
  </netNamedPipeBinding>
...
</appSettings>
...

```

B.3 Web options

This section shows the administrator-configurable options available in the *web.config* file.

Turning the use of the ActiveX control on or off

The 1E Web WakeUp provides an ActiveX control used to retrieve the local computer name for the purposes of registration. When this ActiveX control is not used 1E Web WakeUp will use a reverse DNS lookup that relies on a correctly configured and up-to-date DNS environment. The ActiveX alternative is more reliable, accurate and efficient.

In certain circumstances you may want to not install the ActiveX control, for example 64-bit browsers are not supported by the ActiveX control or you may have pre-determined all the users' registered machines and not want the ActiveX control installed. In these cases you can turn off the installation and use of the ActiveX control. The following table shows the key in the *web.config* file that must be changed to turn this on or off.

Key name	Description
UseActiveX	Set this to "True" to turn on the use of ActiveX. Set this to "False" to turn off the use of ActiveX.

The following configuration file fragment shows an example where the use of ActiveX is turned off.

```
...  
<appSettings>  
...  
<add key="UseActiveX" value="False"/>  
...  
</appSettings>  
...
```

Appendix C Automatically setting the authorization table

The 1E Web WakeUp tables added to the Agility Framework are also supplied with an SQL Server Agent job that can be used to populate the *tbWWU_Authorised_UserMachines* table from a .csv file.

The SQL Server Agent job is called *1E Web WakeUp, Import Authorisation*. Before running it you must first create the source authorization input file and then set the SQL Server Agent job to use the file.

Note: running the SQL Server Agent job will clear any existing data from the authorization table, except for "power user" type entries where the machine name is set to "*".

C.1 The authorization input file format

The input file for the *1E Web WakeUp, Import Authorisation* SQL Server Agent job is a .csv format file where each line in the file should contain the following information in the following format:

<User ID>, <Computer Name>, <Domain Name>

- <User ID> names the user whose authorization is being set. When using Windows authentication <UserID> should be in the format *DomainName\UserName*.
- <Computer Name> provides the netBIOS name of the machine the named user is allowed to wake.
- <Domain Name> denotes the domain the machine is running on.

Note: although the first line of the .csv file must also have three comma separated elements, it is treated as a header line and ignored.

The following shows an example authorization input file with three entries. Notice how the first line, as it is ignored by the import process, can be used as a place to store notes on the file:

```
Revised 9/4/2009 ESA for wwU implementation,,
ACME\rachel,devwks1,ACME
ACME\rachel,devwks2,ACME
ACME\joey,saleswks2,ACME
```

C.2 Setting the authorization input file

To set the SQL Server Agent job to reference a specific authorization file you need to modify the SQL Server Agent job's *Import CSV* step in SQL.

By default the command line for this step is:

```
EXEC [dbo].[spwwU_Import_Authorised_UserMachines] 'C:\CN-DN-PU.csv'
```

You should change the default file referenced at the end of the command line to refer to the path and filename for your .csv file. For example the following sets the SQL Server Agent job *1E Web WakeUp, Import Authorisation* to use *C:\1EWWU\ACME\acme.csv*.

```
EXEC [dbo].[spwwU_Import_Authorised_UserMachines] 'C:\1EWWU\ACME\acme.csv'
```

To edit this job in SQL Server Management Studio you can use the following steps:

1. Navigate to SQL Server Agent/Jobs
2. Right-click on *1E Web WakeUp, Import Authorisation* and select *Properties* from the context menu to open the *Job Properties - 1E Web WakeUp, Import Authorisation* dialog
3. In the *Select a page* column on the left, select *Steps* to display the *Job Step List* in the right pane
4. Select the only job step *Import CSV* and click the *Edit* button to display the *Job Step Properties - Import CSV* dialog
5. In the *Command:* field edit the currently displayed SQL command from:

```
EXEC [dbo].[spwwU_Import_Authorised_UserMachines] 'C:\CN-DN-PU.csv'
```

And change the path\file text at the end to refer to your .csv file. For example the following uses the file `C:\1EWWU\ACME\acme.csv`

```
EXEC [dbo].[spWWU_Import_Authorised_UserMachines] 'C:\1EWWU\ACME\acme.csv'
```

6. Click *OK* to save your changes and close the *Job Step Properties - Import CSV* dialog
7. Click *OK* to close the *Job Properties - 1E Web WakeUp, Import Authorisation* dialog
8. You can then execute the modified job to import the specified data
 - a. Right click on the *SQL Server Agent/Jobs/1E Web WakeUp, Import Authorisation* node and select *Start Job at Step...* item from the context menu
 - b. The job will start running. In the *Start Jobs - <ServerName>* dialog confirm that the *Status* returned at the end is *Success*
 - c. Click *Close* to complete the job execution
9. After execution open the table `tbWWU_Authorised_UserMachines` to confirm the results are as expected

C.3 When the SQL Server Agent job is run

The following steps shows what happens when the SQL Server Agent job is run:

1. Existing entries in the authorisation table are removed, except for "power user" type entries where the machine name is set to "*"
2. The new entries from the input file are added to the authorisation table

Note: one consequence of this sequence is that "power user" type entries where the machine name is set to "*" should only ever be added to the database by hand and not added to the input .csv file as this may result in the import failing due to primary key violations in the database.