

# Upgrading a SIMS Workstation



**CAPITA**  
EDUCATION SERVICES

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

© Capita Business Services Ltd 2007. All rights reserved.  
No part of this publication may be reproduced, photocopied, stored on a retrieval system, translated or transmitted without the express written consent of the publisher.

Capita Doc Ref: WS\_UPG/HBK/150107

Capita Education Services, Franklin Court, Priory Business Park, Cardington, Bedford, MK44 3JZ  
Tel: 01234 838080 Fax: 01234 832082 <http://sims.capitaes.co.uk>

#### **Providing Feedback on Documentation**

We always welcome comments and feedback on the quality of our documentation including online help files and handbooks. If you have any comments, feedback or suggestions regarding the module help file, this handbook (PDF file) or any other aspect of our documentation, please email:

[Publications@capita.co.uk](mailto:Publications@capita.co.uk)

Please ensure that you include the module name, version and aspect of documentation on which you are commenting.

# Table of Contents

<b>Chapter 1: Upgrading a SIMS Workstation.....</b>	<b>1</b>
Introduction.....	1
Important Considerations Before Attempting to Upgrade a SIMS Workstation.....	1
Are Special Access Rights Required to Run a Workstation Upgrade? .....	1
Checking the SIMS.INI File .....	3
Checking Drive Mappings .....	3
Checking the Accuracy of the CONNECT.INI File .....	4
Can My SIMS .net Shortcut Launch SIMS Upgrades? .....	4
Setting System Access Rights for Users (Running SIMSPerm.BAT).....	4
What is SIMSPerm.BAT?.....	5
What Access Rights Does SIMSPerm.BAT Grant? .....	5
What Other Access Rights, Not Provided by SIMSPerm.BAT, are Required? ..	6
When Should I Run SIMSPerm.BAT?.....	6
Who Can Run SIMSPerm.BAT? .....	6
Running SIMSPerm.BAT .....	7
What Will Happen if I Do Not Run SIMSPerm.BAT? .....	8
Upgrading a SIMS Workstation .....	8
<b>Chapter 2: Additional Information.....</b>	<b>9</b>
Using Command Line Scheduler to Automate the SIMS Upgrade Process.....	9
Troubleshooting a SIMS Workstation Upgrade .....	9
Troubleshooting the SIMS.INI File .....	9
Mapping a Drive in Windows Explorer .....	10
Disconnecting a Mapped Drive in Windows Explorer .....	11
Troubleshooting the CONNECT.INI File .....	11
Creating a SIMS .net Shortcut on the Desktop .....	14
What is the Purpose of the Files in the \SIMS\Setups Folder?.....	15
Database Upgrade Files .....	15
AD Provisioning Service Setup Files.....	16
Microsoft SQL Desktop Engine Setup Files .....	16

Nova-T Satellite Setup Files.....	17
AMPARK Setup Files.....	17
SIMS Application Setup Files (Workstation Setup Files) .....	18
SIMS Document Server Setup Files .....	18
SIMS Infrastructure Setup Files .....	18
SIMS Installer Files .....	19
SIMS Manual Setup Files.....	19
SIMS SQL Application Setup Files.....	19
<b>Appendix.....</b>	<b>21</b>
Overview of SOLUS .....	21
Using the SOLUS 2 Wizard to Upgrade your SIMS System .....	22
What Happens During a Download from the SOLUS Website?.....	22
Using Scheduled Tasks to Automate the SOLUS Download Process .....	24
<b>Glossary.....</b>	<b>29</b>
<b>Index.....</b>	<b>31</b>

## Chapter 1: Upgrading a SIMS Workstation

This chapter contains:

---

Introduction.....	1
Important Considerations Before Attempting to Upgrade a SIMS Workstation.....	1
Setting System Access Rights for Users (Running SIMSPerm.BAT).....	4
Upgrading a SIMS Workstation .....	8

---

### Introduction

This document is designed to provide advice to the SIMS system manager, regarding the management of your SIMS workstation upgrades. It includes information on how to set up the required user permissions on the school network in preparation for a workstation upgrade, the workstation upgrade itself and how the whole process can be automated to reduce the amount of manual effort required to carry out this process.

For information on the initial download of the updated SIMS software from the SOLUS website, please refer to the *Appendix on page 21*.

### Important Considerations Before Attempting to Upgrade a SIMS Workstation

Once the download of the latest SIMS software from the SOLUS website has completed successfully and the SIMS SQL database has been upgraded, the next step in the process is to deploy these updates to each workstation that runs SIMS.

Once the appropriate preparation work has taken place, the SIMS workstation upgrade should happen automatically on logging into SIMS .net.

Before deploying these updates, you are advised to read the following section, which includes important information regarding workstation upgrades. It also includes information on checks you should carry out prior to performing workstation upgrades.

Should you encounter any problems during the workstation upgrade process, please refer to *Troubleshooting on page 9*.

### Are Special Access Rights Required to Run a Workstation Upgrade?

A number of special system access rights are required in order to perform a workstation upgrade, some of which might already have been assigned to school network users.

As the system manager, you would have already granted access rights to users for certain areas of the network and to certain areas of a workstation. In order to perform a workstation upgrade, each SIMS user needs high level access rights to some of the system folders on the workstation, together with access to edit the registry of a SIMS workstation.

To reduce the need to grant high levels of access to users on a workstation, a batch file has been provided which grants specific rights to the required SIMS files and folders, and to individual areas of the registry. Using this batch file to grant permissions reduces the need to grant system access rights manually, whilst still maintaining any access restrictions you have already put in place.

The batch file, named `SIMSPerm.BAT`, must be run for *each* SIMS user on *each* SIMS workstation that they are likely to use. For example, if Anita Abell needs to use SIMS, she generally uses the SIMS workstation situated in the staff room. From time to time, she needs to access SIMS using Jane's workstation in the school office. To ensure that Anita can log into SIMS and perform workstation upgrades on both SIMS workstations when required, `SIMSPerm.BAT` must be run for Anita on both workstations (and any other workstation from which she is likely to access SIMS). `SIMS.Perm.BAT` must also be run for Anita's colleague Jane, to ensure that she too can run SIMS on her workstation in the school office. Alternatively, `SIMSPerm.BAT` can be run once on each workstation with a parameter which allows it to be run for a group of network users.

---

**IMPORTANT NOTES:** *SIMSPerm.BAT* does not need to be run prior to each SIMS upgrade. If you add a system group name as a parameter when running *SIMSPerm.BAT*, you will have to run this file only once.

*If a user is added to the system group after SIMSPerm.BAT has been run, they will inherit the permissions already granted to the existing members of the system group. Therefore it is not necessary to re-run SIMSPerm.BAT.*

*If you run SIMSPerm.BAT for individual users, SIMSPerm.BAT must be run on any workstation that that user is likely to use.*

*SIMSPerm.BAT MUST be run by a system user with Local Administrator access rights. No other user can run SIMSPerm.BAT.*

---

If you are uncertain whether or not you need to run `SIMSPerm.BAT` for a particular user or group of users, ask the user(s) to log into SIMS .net following an upgrade. If the workstation setup completes successfully, the user(s) has the appropriate permissions. If the workstation upgrade fails to complete successfully, you are advised to run `SIMSPerm.BAT` for that user.

For more information on how `SIMSPerm.BAT` can benefit your school, together with a list of the permissions it grants to users, please refer to *Setting System Access Rights for Users* on page 4.

## Checking the SIMS.INI File

SIMS.INI contains important information relating to the SIMS system, both on the workstation and on the server. This file can be found in the \WINDOWS folder on the workstation.

---

*NOTE: If SIMS has been installed successfully with the correct parameters defined, the SIMS.INI file should be correct. However if the file has been edited manually, an error might have been introduced to the file.*

---

Prior to upgrading a SIMS workstation, you should check a number of settings in the SIMS.INI file.

Firstly, check that the file contains a reference to the local SIMS .net folder. The line should look similar to the following:

**SIMSDotNetDirectory=C:\Program Files\SIMS\SIMS .net**

Next, check that the file contains a reference to the SIMS Setups folder. The line should look similar to the following:

**SIMSSetupsDirectory=S:\SIMS\Setups**

Finally, check that the file contains a reference to the SIMS folder. The line should look similar to the following:

**SIMSDirectory=S:\SIMS**

This is the location of the SIMS installation on the server.

For more information on resolving issues with the SIMS.INI file, please refer to *Troubleshooting the SIMS.INI File on page 9*.

## Checking Drive Mappings

If SIMS is accessed from a server (which under most circumstances you will be, unless you are working in standalone mode), it is imperative that the appropriate drive mappings have been set up for each user on each workstation that accesses SIMS.

Drive mapping details are stored in a user profile, so if one user on a workstation has a mapping set up to the SIMS server, this does not necessarily mean that other users on that workstation will also have drive mappings set up.

Drive mappings can be set up in Windows Explorer. In the panel on the left-hand side, browse the list of drives for the SIMS server (e.g. S:\). If a drive mapping does not exist for each user that runs SIMS on that workstation, one must set one up prior to opening SIMS .net.

For more information on mapping network drives and disconnecting mapped drives, please refer to *Mapping a Drive in Windows Explorer on page 10* and *Disconnecting a Mapped Drive in Windows Explorer on page 11*, respectively.

## Checking the Accuracy of the CONNECT.INI File

---

**IMPORTANT NOTE:** *It is essential that ONLY the SIMS system manager edits this file. Incorrect use of this file could lead to SIMS .net failing to work on a workstation.*

---

The CONNECT.INI file tells SIMS .net which SIMS SQL database to use with the application. It is *essential* that the correct database is referenced in the CONNECT.INI file, to ensure that the current, live database is being used to store accurate and up-to-date data. For example, if your live school database is named **SIMS**, you must ensure that this database name is referenced in CONNECT.INI. If an incorrect database name is entered, the user will be viewing or editing data in the incorrect database (i.e. *not* the live database).

For more information on resolving issues with the CONNECT.INI file, please refer to *Troubleshooting the CONNECT.INI File on page 11*, *Understanding a 'Master' CONNECT.INI File on page 13* and *Understanding a 'Redirect' CONNECT.INI File on page 13*.

## Can My SIMS .net Shortcut Launch SIMS Upgrades?

Once SOLUS has been used to download the latest upgrades to the SIMS system, individual SIMS workstations are upgraded automatically when SIMS .net is opened for the first time following the download.

It is no longer necessary to open SIMS .net by selecting **Start | All Programs | SIMS Applications | SIMS .net** from the Taskbar. Instead, with a minor alteration to the SIMS .net shortcut, it is also possible to launch SIMS .net immediately after an upgrade by double-clicking on the SIMS .net shortcut icon on the Windows desktop.

To take advantage of this simplified method of opening SIMS .net after an upgrade, a reference to SIMSLOAD must be included in the shortcut. The reference to SIMSLOAD is vital, as this forces the system to look at a file named SIMSLOAD.INI, which then determines whether or not any upgrade files need to be run. Without the reference to SIMSLOAD.INI, the shortcut will not launch any upgrades.

For more information on creating a SIMS .net desktop shortcut that can trigger a workstation upgrade, please refer to *Creating a SIMS .net Shortcut on the Desktop on page 14*.

## Setting System Access Rights for Users (Running SIMSPerm.BAT)

In order to upgrade each workstation on the SIMS system, the user logging into SIMS .net *must* be granted Local Administrator rights.

---

*NOTE: Workstation upgrades are carried out the first time a user logs into SIMS .net on a workstation after a SOLUS download.*

---

Not all system managers will want to grant high level workstation access rights to users. Instead, they can run `SIMSPerm.BAT` for a user or group of users in order to grant the necessary access rights to run a workstation upgrade.

---

*NOTE: `SIMSPerm.BAT` grants access rights to an individual workstation ONLY. It does NOT edit a user's network access rights, nor does it edit a user's Permissions in `SIMS.net`.*

---

## What is `SIMSPerm.BAT`?

`SIMSPerm.BAT` is a batch file which can be run so that the workstation can be upgraded without the need for logging on as a Local Administrator. It grants a number of access rights to an individual system user or a group of system users. The access rights granted to a user once `SIMSPerm.BAT` has been run are equivalent to those of a *Power User*, to the parts of the local workstation required by SIMS.

For simplicity, the system manager might want to run the batch file with the `/s` parameter for the system groups, to reduce the number of times it has to be run per user (assuming that the appropriate system groups have been set up).

---

**IMPORTANT NOTE:** *Running this batch file means that it will not be necessary to log in as a Local Administrator on each workstation in order to upgrade SIMS. However, it does grant further workstation access rights to users. After the access rights have been granted, inappropriate use of the system would make it possible to damage the registry. Therefore, system managers must make the choice between system automation and security.*

---

## What Access Rights Does `SIMSPerm.BAT` Grant?

The following lists indicate the access rights that will be granted to each user once the batch file has been run.

### Access Rights to the Registry

Full access is granted to the following areas of the Registry:

`HKEY_LOCAL_MACHINE\SOFTWARE\Classes`

`HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID`

`HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface`

`HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib`

### Access Rights to Folders

Grants *full* access to the local installation of SIMS  
`\Program Files\SIMS` folder and subfolders.

Grants *full* access to a **SIMS Applications Folder** in the start menu  
`%ALLUSERSPROFILE%\Start Menu\Programs\SIMS Applications\.`

### Access Rights to Files

Grants *full* access to the SIMS.INI file  
%WINDIR%\SIMS.INI.

### What Other Access Rights, Not Provided by SIMSPerm.BAT, are Required?

Full access must also be granted to S:\SIMS (where S:\SIMS is the location of your SIMS installation on the network) for those SIMS users who still require access to the following areas:

- FMS
- DENI
- System Manager for DOS
- Launcher
- Personnel 5
- archived STAR students
- Web Integrator for Parents Gateway.

---

**IMPORTANT NOTE:** *SIMSPerm.BAT* provides access rights to certain areas of the workstation only, and therefore does not affect network access rights. Access rights to the S:\SIMS folder must be assigned manually by the network manager.

---

### When Should I Run SIMSPerm.BAT?

It should be run after SIMS .net has been installed. It will need to be run for *each* user (or each group of users) on *each* workstation from which they will be running SIMS, in order to set up the permissions. Once it has been run for each user or group of users, the workstations will be upgraded automatically each time a user logs into SIMS .net and an upgrade is available.

---

*NOTE: If you are using either Microsoft SMS or RM CC3 deployment tools, the workstations can be set up remotely. However, running SIMSPerm.BAT on each SIMS workstation is the recommended method to achieve the appropriate levels of access rights for SIMS users, so that SIMS can be upgraded after deployment.*

---

### Who Can Run SIMSPerm.BAT?

Only Local Administrators can run SIMSPerm.BAT.

---

## Running SIMSPerm.BAT

---

*NOTE: The following instructions should be performed on any machine running a Windows NT based operating system (Windows 2000, Windows XP, etc.) that will be used as a SIMS workstation. As the SIMS SQL server will be updated by an administrator, SIMSPerm.BAT need not be run on this machine. **SIMSPerm.BAT should NOT be run on Windows 98 or NT4, as these operating systems are no longer supported.***

---

1. Select **Start** | **Run** from the Taskbar to open the **Run** dialog.

2. In the **Open** field, enter:

**S:\SIMS\Setups\SIMSPerm.BAT**

(where *S:* is the location of the SIMS file server)

3. SIMSPerm.BAT requires an additional parameter in order to identify the system user or group of system users for whom you wish to set access rights.

To assign the appropriate access rights to a single user, enter:

**S:\SIMS\Setups\SIMSPerm.BAT <Domain Name>\<System User Name>**

or

**S:\SIMS\Setups\SIMSPerm.BAT <System User Name>**

(removing the Domain Name from the parameter will force the batch file to search for a locally stored user.)

To assign the appropriate access rights to a group of users, enter:

**S:\SIMS\Setups\SIMSPerm.BAT <Domain Name>\<SystemUserGroup>**

---

*NOTE: If <SystemUserGroup> contains spaces this should be enclosed in double quotes, e.g. <"System User Group">.*

---

4. Click the **OK** button.

When the batch file has been run successfully, a user can view their new access rights (to the \Program Files\SIMS folder and subfolders and the %ALLUSERSPROFILE%\Start Menu\Programs\SIMS Applications\ folder, for example) by right-clicking on the appropriate folder in Windows Explorer and selecting **Properties** to open the <Folder Name> **Properties** dialog, then selecting the **Security** page.

---

**IMPORTANT NOTE:** Running SIMSPerm.BAT without any parameters (i.e. double-clicking the file in Windows Explorer) will NOT affect a user's system access rights.

---

## What Will Happen if I Do Not Run SIMSPerm.BAT?

Running SIMSPerm.BAT is the easiest and the recommended way of achieving the appropriate level of access rights in order to run SIMS .net upgrades automatically. If you do *not* run SIMSPerm.BAT, you will have to ensure that every user has been granted Local Administrator rights before they can upgrade SIMS .net automatically or a user with Local Administrator rights will need to visit each workstation in order to run the upgrade.

However, if the system manager already grants high level system access rights to their users, they might not have to run SIMSPerm.BAT.

## Upgrading a SIMS Workstation

---

*NOTES: SIMS .net must be run on any workstation new to SIMS prior to attempting to run Web Integrator on that workstation.*

*Lesson Monitor can be opened either from the **Start** menu on the Taskbar, or from a desktop shortcut icon. If you wish to use a desktop shortcut icon, you must ensure that the **Target** line includes a reference to SIMSLOAD. For more information, please refer to Can My SIMS .net Shortcut Launch SIMS Upgrades? on page 4 and Creating a SIMS .net Shortcut on the Desktop on page 14.*

---

When a user logs into SIMS .net for the first time following an upgrade, the workstation will be updated automatically.

---

*NOTES: This process must be carried out on ALL workstations.*

*It does not require an administrative visit as long as the user has been given adequate permissions for the update to occur. For more information, please refer to Setting System Access Rights for Users on page 4.*

---

1. Select **Start | Programs | SIMS Applications | SIMS .net** to open the module.
- 

*NOTE: It is also possible to launch SIMS .net immediately after an upgrade by double-clicking on the SIMS .net shortcut icon on the Windows desktop. However, this will only work if the shortcut has been amended to reference SIMSLOAD. For more information, please refer to Can My SIMS .net Shortcut Launch SIMS Upgrades? on page 4.*

---

2. The update will be performed automatically. Once the update has completed, SIMS .net can be opened and used.

## Chapter 2: Additional Information

This chapter contains:

---

Using Command Line Scheduler to Automate the SIMS Upgrade Process.....	9
Troubleshooting a SIMS Workstation Upgrade .....	9
What is the Purpose of the Files in the \SIMS\Setups Folder?.....	15

---

### Using Command Line Scheduler to Automate the SIMS Upgrade Process

For instructions on how to set up the command line scheduler, which can be used to automate the SIMS Upgrade process, please refer to the *Command Line Use of SIMS Installers* document. This document is available from the SOLUS web site, [www.capitasolus.co.uk](http://www.capitasolus.co.uk), by clicking the **SOLUS 2 Documentation** link and clicking the document title.

### Troubleshooting a SIMS Workstation Upgrade

A SIMS workstation upgrade might fail for a number of reasons. This section of the guide discusses the possible reasons for workstation upgrade failure and possible resolutions.

#### Troubleshooting the SIMS.INI File

A number of settings in the `SIMS.INI` file should be checked.

Firstly, check that the file contains a reference to the local SIMS .net folder. The line should look similar to the following:

```
SIMSDotNetDirectory=C:\Program Files\SIMS\SIMS .net
```

If this line is missing, it will not be possible to download the upgrade files from SOLUS. If this line points to an incorrect location, it could cause SIMS .net to build its upgrade files in the wrong location, causing the upgrade to fail as there will be no upgrade files in the correct location. If this line is missing, or if it points to an incorrect location, workstation upgrade files downloaded from SOLUS will not be applied to the workstation.

Next, check that the file contains a reference to the SIMS Setups folder. The line should look similar to the following:

```
SIMSSetupsDirectory=S:\SIMS\Setups
```

SIMS .net will attempt to locate upgrade files in this location. If the appropriate upgrade files cannot be found in this location, SIMS .net will not be upgraded. If you have recently moved the SIMS server, the drive mapping might also have changed. In this case, ensure that the drive letter to which this line refers still exists. Alternatively, you can use a UNC path, e.g. \\SERVERNAME\SIMS\SETUPS. The main benefit of using a UNC path is that, unlike drive mappings set up in Windows Explorer, this type of path cannot be removed accidentally. As it does not rely on a drive mapping, you will not need to change the path if any system drive mappings change.

---

**IMPORTANT NOTES:** *If you use drive mappings instead of a UNC path, and the drive mapping to the SIMS server changes (e.g. from S: \ to X: \), the change of drive mapping will have to be reflected in this line of the SIMS.INI file on each SIMS workstation.*

*A UNC path describes the location of a volume, folder or a file. The format for a UNC path is \\SERVER\VOLUME\FOLDER\FILE and is not case sensitive.*

---

Finally, check that the file contains a reference to the SIMS folder. The line should look similar to the following:

**SIMSDirectory=S:\SIMS**

This is the location of the SIMS installation on the server.

---

*NOTE: Where any of the examples in this section refer to the SIMS server, e.g. S:\SIMS, you are advised to check that a drive mapping has been set up. If the drive mapping is not set up, it will not be possible to obtain SIMS .net upgrades or to run SIMS .net. For more information on mapping drives, please refer to Checking Drive Mappings on page 3. Alternatively, you can use UNC paths.*

---

## Mapping a Drive in Windows Explorer

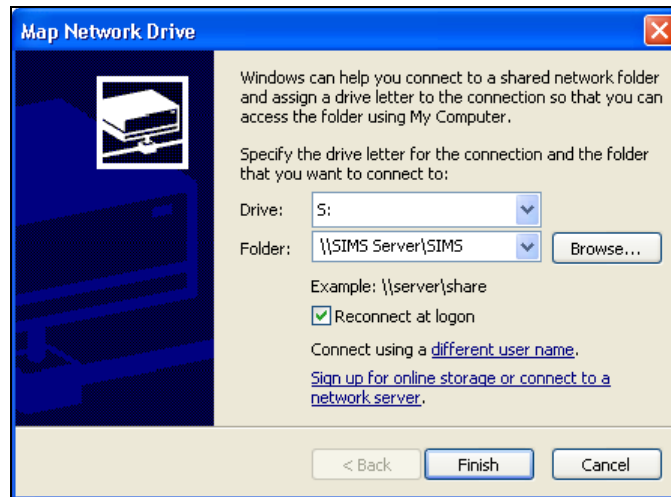
If a drive mapping to the SIMS server has not already been created, it must be set up before opening SIMS .net and before running workstation upgrades.

This must be done for each user who will access SIMS on this workstation, because drive mappings are made on a per user basis and the information is stored in the user profile.

1. Select **Start | All Programs | Accessories | Windows Explorer** from the Taskbar to open Windows® Explorer.
2. In the left-hand panel, check the list of mapped network drives to see if a mapping already exists for the SIMS server, e.g. **SIMS on 'SIMS Server' (S:)**.

If a mapping does not already exist, it must be created. If it does exist, no further work is required.

3. Select **Tools | Map Network Drive** to open the **Map Network Drive** dialog.



4. Select the drive letter you wish to use as the mapped drive from the **Drive** drop-down list.
5. Select the location of the SIMS folder on the server by clicking the **Browse** button to open the **Browse for Folder** dialog.
6. Navigate to the required location and click the **OK** button.
7. Ensure that the **Reconnect at logon** check box is selected so that access to this drive is available whenever a user logs into the network.
8. Once you are sure that the information entered is correct, click the **Finish** button.



#### **More information:**

*Checking the SIMS.INI File on page 3*

*Checking the Accuracy of the CONNECT.INI File on page 4*

### **Disconnecting a Mapped Drive in Windows Explorer**

If a mapped drive has been added in error, or if the incorrect drive has been mapped, the mapping can be removed.

1. Select **Start | All Programs | Accessories | Windows Explorer** from the Taskbar to open Windows® Explorer.
2. Select **Tools | Disconnect Network Drive** to open the **Disconnect Network Drives** dialog.
3. Highlight the network drive whose mapping you wish to remove and click the **OK** button.

The network drive will no longer be displayed as a mapping in Windows Explorer.

### **Troubleshooting the CONNECT.INI File**

The CONNECT.INI file can be found in two locations:

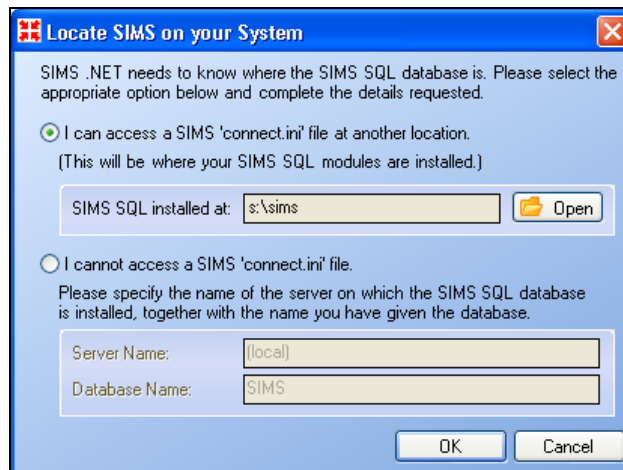
- in the \SIMS folder on the SIMS server; and
- in the \Program Files\SIMS\SIMS .net folder on a workstation.

These files are created on installation of SIMS .net and when a workstation is being prepared to run SIMS for the first time, respectively. The version in the \SIMS folder on the SIMS server should *always* point to the SIMS server and the live database name.

The version in the \Program Files\SIMS\SIMS .net folder can point to the SIMS server and the live database name, or it can be 'redirected' so that it uses the information stored in the master CONNECT .INI file on the SIMS server. The main benefit of using a 'redirect' CONNECT .INI on a workstation, is that if the server name or database name changes, it will not be necessary to make any changes to the CONNECT .INI file on every SIMS workstation.

Prior to running a SIMS workstation upgrade, please ensure that the CONNECT .INI file is present, accessible and accurate.

If for any reason the CONNECT .INI file is not available on the workstation when logging into SIMS .net (e.g. it has been renamed or deleted), the following dialog will be displayed. This dialog will also be displayed the first time anybody logs into SIMS .net on that workstation.



Select the **I can access a SIMS 'connect.ini' file at another location** radio button if you wish to redirect the CONNECT .INI on the workstation to look at the master CONNECT .INI on the SIMS server. Click the **Open** button to browse for the correct version of the CONNECT .INI file (e.g. S : \SIMS) and click the **OK** button.

Select the **I cannot access a SIMS 'connect.ini' file** radio button if you cannot locate or access the CONNECT .INI on the SIMS server. Selecting this radio button will create a 'master' CONNECT .INI on the workstation (e.g. C : \Program Files\SIMS\SIMS .net). Enter the **Server Name** and **Database Name** and click the **OK** button.

### Understanding a 'Master' CONNECT.INI File

In the C:\Program Files\SIMS\SIMS .net folder (or in the \SIMS folder on the SIMS server), double-click the CONNECT.INI file to open a text file containing text similar to the following:

```
[SIMSConnection]
ServerName=Gabbey\SIMS
DatabaseName=SIMS
```

This type of CONNECT.INI file is found mainly in the \SIMS folder on the server. It contains details of the name and location of the SIMS server, together with the name of the SIMS SQL database. By default on installation, each workstation will be allocated another type of CONNECT.INI file (a 'redirect' file) which redirects to the CONNECT.INI file on the server, to facilitate maintenance of the system. For more information on a CONNECT.INI which redirects to the file on the server, please refer to *Understanding a 'Redirect' CONNECT.INI File on page 13*.

This type of CONNECT.INI file can also be used on each SIMS workstation. However, using this type of CONNECT.INI on a workstation will mean that you will have to visit each workstation if either the SIMS server name or the database name changes.

---

**IMPORTANT NOTE:** *If the drive letter to which SIMS is mapped changes, it will still be necessary to visit each workstation to edit the drive letter in the local CONNECT.INI file.*

---

If the content of the CONNECT.INI file looks similar to the previous example, check with the system manager that the CONNECT.INI file references the correct server name and database.

This would be particularly important if the SIMS database has recently been moved from one server to another, or if the database name has been changed, and the CONNECT.INI file has not yet been updated with the new details. Attempting to run SIMS .net whilst the CONNECT.INI file references a non-existent server or an invalid database will cause an error message to be displayed.

---

**IMPORTANT NOTE:** *The contents of the CONNECT.INI will vary from school to school. The example shown here is a master CONNECT.INI file, but additional lines might be displayed if, for example, your school uses Teachers' Desktop functionality.*

---

### Understanding a 'Redirect' CONNECT.INI File

In the C:\Program Files\SIMS\SIMS .net folder, double-click the CONNECT.INI file to open a text file containing text similar to the following:

```
[SIMSConnection]
Redirect=S:\SIMS
```

This type of `CONNECT.INI` file is widely used on a SIMS workstation because it requires very little maintenance. It acts as a 'holding' `CONNECT.INI` file, as its sole function is to point to a centrally stored `CONNECT.INI` file and access the server and database details held therein.

The redirect in the file on the workstation ensures that any change made to the `CONNECT.INI` on the server is picked up on each workstation. The main benefit of using this type of `CONNECT.INI` file is if, for example, the SIMS database is moved to another server. Instead of visiting each SIMS workstation and editing the server name in each `CONNECT.INI`, only the `CONNECT.INI` on the server, which is shared by all SIMS users, needs to be changed.

You would generally use this type of `CONNECT.INI` file on a workstation to reduce the need to visit each workstation if any of the server details change.

---

**IMPORTANT NOTE:** *If the drive letter to which SIMS is mapped changes, it will still be necessary to visit each workstation to edit the drive letter in the local `CONNECT.INI` file.*

---

If the content of the `CONNECT.INI` file looks similar to the previous example, please carry out the following steps:

- Check that the path displayed on the **Redirect** line is accessible from the workstation (e.g. `S:\SIMS`, where `S:\` is the location of the SIMS server). This can be achieved by opening Windows Explorer and checking whether or not a drive mapping exists to that drive.

If a drive mapping exists, it should be possible to access the `CONNECT.INI` file in that location. If it does not exist, you will have to set up a drive mapping. For more information, please refer to *Mapping a Drive in Windows Explorer on page 10*.

The path displayed on the **Redirect** line will contain another `CONNECT.INI` file, which will hold details of the server name and database name.

- Check that the `CONNECT.INI` file exists in the path specified on the **Redirect** line, by searching for it in Windows Explorer.
- Check with the system manager that the `CONNECT.INI` file on the server references the correct database.

## Creating a SIMS .net Shortcut on the Desktop

Shortcuts can be created in various ways. SIMS .net shortcuts will only trigger the upgrade if they have been created in a certain way. To check whether or not the SIMS .net shortcut on the desktop will trigger an upgrade:

1. Right-click on the shortcut and select **Properties**.
2. On the **Shortcut** page, check that the following text (including speech marks) is entered in the **Target** field:

```
"C:\Program Files\SIMS\SIMS .net\SIMSLoad.exe" "C:\Program Files\SIMS\SIMS .net\Pulsar.exe"
```

If all of this text is present, the shortcut will trigger the upgrade. If some of the text is missing, the shortcut should be deleted and a new one should be created:

1. Highlight the shortcut and press the **Delete** key. Confirm that you wish to delete the shortcut.
2. Select **Start | Programs | SIMS Applications** from the Taskbar.
3. Right-click on **SIMS .net** and drag it to the desktop.
4. Select **Copy Here** from the pop-up menu.

If you right-click on the newly created SIMS .net shortcut and select **Properties**, you will see that the **Target** field on the **Shortcut** page contains the correct text.

## What is the Purpose of the Files in the \SIMS\Setups Folder?

When you download the latest version of SIMS using the SOLUS 2 wizard, the downloaded files are posted to the \SIMS\Setups folder on the SIMS server. The quantity of files that need to be downloaded is directly related to the length of time since your last download; generally speaking, the longer the amount of time, the larger the number of files. The \SIMS\Setups folder therefore contains all of the files downloaded from the SOLUS server.

This section discusses the different types of files that can be found in the \SIMS\Setups folder and discusses their importance to the SIMS upgrade process.

### Database Upgrade Files

Database upgrade files are applied to the database in order to update it in line with the latest version of the SIMS software. Changes that can be applied via a database upgrade include the addition of data fields into a database table, the addition of database tables into the SIMS SQL database and the amendment of system permissions.

After downloading the latest version of SIMS via SOLUS, the system upgrades the database automatically. This is achieved using a utility called **dbUpgrade** (which can be found in the \Binn folder on the SQL server), which looks in the \SIMS\Setups folder for upgrade files. Each step of a database upgrade requires a pair of files to be present in order to be run successfully. These files are named as follows: (this example assumes the database version prior to upgrade is **3.61.331**).

3.61.331 to 3.61.332 Upgrade.zip

3.61.331 to 3.61.332 Upgrade.zip.signature

where **3.61.331** is the current database version and **3.61.332** will be the version of the database once the upgrade has been applied.

The .ZIP file contains the SQL script to be run and the .signature file verifies whether or not you can run the upgrade on your database.

**IMPORTANT NOTES:** The pair of `.ZIP` and `.signature` files is required for a database upgrade to complete successfully. If either is missing when running `dbUpgrade`, the database upgrade will fail.

Any number of database upgrades can be applied at the same time. `dbUpgrade` runs all valid upgrade files automatically.

---

## AD Provisioning Service Setup Files

The Active Directory Provisioning Service is the Windows Service which runs on the Data Tier, or any other machine connected to the Data Tier and the Application Tier server. The role of this service is to check the database for any users who are yet to be provisioned for Active Directory and Exchange Server in relation to SIMS Learning Gateway). This service is also responsible for synchronising the distribution lists (memberships) of the users who have already been provisioned.

The **Active Directory Provisioning Service** wizard is upgraded automatically when SIMS `.net` is updated via SOLUS.

Each AD Provisioning Service Setup upgrade requires two pairs of files to be present in order to be run successfully. The first pair of files contains the updated wizard, and the files are named as follows: (this example assumes the setup file version currently being run is **1.548.3.0**).

`ADProvisioningServiceSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe`

`ADProvisioningServiceSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe.signature`

where **1.548.3.0** is the current AD Provisioning Service setup file version and **1.548.4.0** will be the version once the upgrade has been applied.

The second pair of files runs the AD Provisioning Service upgrade, and the files are named as follows:

`ADProvisioningServiceSetup.exe`

`ADProvisioningServiceSetup.exe.signature`

## Microsoft SQL Desktop Engine Setup Files

The following files install Microsoft SQL 2000 Desktop Engine Service Pack 3:

`MSSQL2000DeskSP3.exe`

`MSSQL2000DeskSP3.exe.signature`

The following files upgrade Microsoft SQL 2000 Desktop Engine to Service Pack 3:

`MSSQL2000DeskUpgradeSP3.exe`

`MSSQL2000DeskUpgradeSP3.exe.signature`

## Nova-T Satellite Setup Files

These files provide users with the opportunity of working in Nova-T on a standalone PC, away from the school environment.

Each Nova Satellite upgrade requires two pairs of files to be present in order to be run successfully. The first pair of files contains the Nova Satellite upgrade files, and the files are named as follows: (this example assumes the SQL Application setup file version currently being run is **1.546.8.0**).

```
NovaSatelliteSetup 1.546.8.0 to 1.546.10.0 Upgrade.exe
```

```
NovaSatelliteSetup 1.546.8.0 to 1.546.10.0  
Upgrade.exe.signature
```

where **1.546.8.0** is the current SQL Application setup file version and **1.546.10.0** will be the version once the upgrade has been applied.

The second pair of files runs the SQL Application upgrade, and the files are named as follows:

```
NovaSatelliteSetup.exe
```

```
NovaSatelliteSetup.exe.signature
```

## AMPARK Setup Files

The Assessment Manager and Performance Analysis Resource Kit (AMPARK) provides users with a set of resources, predominantly to assist with the Key Stage, Review and Target setting processes. This kit is updated periodically and is upgraded as part of the SIMS upgrade.

Each AMPARK upgrade requires two pairs of files to be present in order to be run successfully. The first pair of files contains the data, and the files are named as follows: (this example assumes the AMPARK setup file version currently being run is **1.542.1.0**).

```
SIMSAMPARKSetup 1.542.1.0 to 1.544.1.0 Upgrade.exe
```

```
SIMSAMPARKSetup 1.542.1.0 to 1.544.1.0 Upgrade.exe.signature
```

where **1.542.1.0** is the current AMPARK setup file version and **1.544.1.0** will be the version once the upgrade has been applied.

The second pair of files runs the AMPARK upgrade, and the files are named as follows:

```
SIMSAMPARKSetup.exe
```

```
SIMSAMPARKSetup.exe.signature
```

## SIMS Application Setup Files (Workstation Setup Files)

These files provide users with updates to the files on their local SIMS .net folder (typically `\Program Files\SIMS\SIMS .net` on the workstation), including the updated SIMS .net executable (`Pulsar.EXE`), updated .DLL files, updated help files (.CHM and .HLP files) and other executable files. Other common .EXE files are also copied to the `\SIMS` folder on the SIMS server. These files are updated automatically the first time a user logs into SIMS .net a workstation after an upgrade.

Each workstation upgrade requires two pairs of files to be present in order to be run successfully. The first pair of files contains the data, and the files are named as follows: (this example assumes the Application setup file version currently being run is **1.548.3.0**).

```
SIMSApplicationSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe
```

```
SIMSApplicationSetup 1.548.3.0 to 1.548.4.0  
Upgrade.exe.signature
```

where **1.548.3.0** is the current Application setup file version and **1.548.4.0** will be the version once the upgrade has been applied.

The second pair of files runs the Application upgrade, and the files are named as follows:

```
SIMSApplicationSetup.exe
```

```
SIMSApplicationSetup.exe.signature
```

## SIMS Document Server Setup Files

These files provide users with updates to the SIMS Document Server. These files are typically copied to the Document Server Application folder.

Each Document Server upgrade requires a pair of files to be present in order to be run successfully. These files provide a program which runs the Document Server update, and are named as follows:

```
SIMSDocumentServerSetup.exe
```

```
SIMSDocumentServerSetup.exe.signature
```

## SIMS Infrastructure Setup Files

These files provide all the technologies required in order for SIMS to function correctly, but are not authored by Capita Education Services. Local Administrator access rights on the workstation are required to run this setup process.

Each Infrastructure upgrade requires a pair of files to be present in order to be run successfully. These files provide a program which runs the Infrastructure update, and are named as follows.

```
SIMSInfrastructureSetup.exe
```

```
SIMSInfrastructureSetup.exe.signature
```

## SIMS Installer Files

This application provides a user-friendly method of selecting which installation routines to run on a server or a workstation. The options are to use a PC as a SIMS SQL Server, as a SIMS Document Server and/or as a SIMS workstation.

The options selected from this application determine the upgrades that will be carried out on a PC. The files are named as follows:

`SIMSINST.exe`

`SIMSINST.exe.signature`

## SIMS Manual Setup Files

These files provide users with updated SIMS .net user handbooks (.PDF), which are copied to their local SIMS .net folder (typically `\Program Files\SIMS\SIMS .net` on the workstation) and the `\SIMS` folder on the SIMS server. These files are updated automatically as part of the SIMS upgrade.

Each handbook upgrade requires two pairs of files to be present in order to be run successfully. The first pair of files contains the handbooks, and the files are named as follows: (this example assumes the Handbook setup file version currently being run is **1.548.3.0**).

`SIMSManualSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe`

`SIMSManualSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe.signature`

where **1.548.3.0** is the current Handbook setup file version and **1.548.4.0** will be the version once the upgrade has been applied.

The second pair of files runs the Handbook upgrade, and the files are named as follows:

`SIMSManualSetup.exe`

`SIMSManualSetup.exe.signature`

## SIMS SQL Application Setup Files

These files provide users with updates to the SIMS SQL back-end programs, such as `dbUpgrade`, `dbAttach` and various .DLL files. These files are typically copied to the `MSSQL\Binn` folder on the SIMS SQL server and are updated automatically as part of the SIMS upgrade.

Each SQL application upgrade requires two pairs of files to be present in order to be run successfully. The first pair of files contains the SQL applications, and the files are named as follows: (this example assumes the SQL Application setup file version currently being run is **1.548.3.0**).

`SIMSSQLApplicationSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe`

`SIMSSQLApplicationSetup 1.548.3.0 to 1.548.4.0 Upgrade.exe.signature`

where **1.548.3.0** is the current SQL Application setup file version and **1.548.4.0** will be the version once the upgrade has been applied.

The second pair of files runs the SQL Application upgrade, and the files are named as follows:

`SIMSSQLApplicationSetup.exe`

`SIMSSQLApplicationSetup.exe.signature`

## Appendix

This chapter contains:

---

Overview of SOLUS .....	21
Using the SOLUS 2 Wizard to Upgrade your SIMS System .....	22
What Happens During a Download from the SOLUS Website? .....	22
Using Scheduled Tasks to Automate the SOLUS Download Process .....	24

---

### Overview of SOLUS

SOLUS 2 is used to download files (including upgrades and patches) provided by Capita Education Services to your SIMS server.

In early 2004, Capita Education Services began releasing upgrades and patches online by means of the **SIMS Online Update Service (SOLUS)**. This has simplified the upgrade and patching process, removing the requirement for CDs and ensuring updates can be applied as soon as they are available.

The **SOLUS 2 wizard** simplifies the process further. It works in conjunction with CES Schools Manager, which is used by LAs and LSUs, so that they can manage the distribution of software to schools. The LAs/LSUs can ensure that only the relevant updates are available for each establishment. Schools will receive email notification of when to run the SOLUS 2 wizard. Once an update has been downloaded and applied by a school, the SOLUS 2 wizard feeds this information back to the LA/LSU automatically, so that support units are kept up-to-date with what version a school is running.

The SOLUS facility has been designed to be easy to use, secure and not requiring extensive technical knowledge. It enables users to upgrade all SIMS .net components, including the database and applications.

The SOLUS 2 wizard exchanges the following data with Capita ES:

- The school number stored in the SQL database.
- The postcode of the school.
- The version of the SIMS database.
- The version of the FMS database.

The SOLUS 2 wizard will also inform Capita ES and the LA/LSU of the following events:

- The school has completed a download.

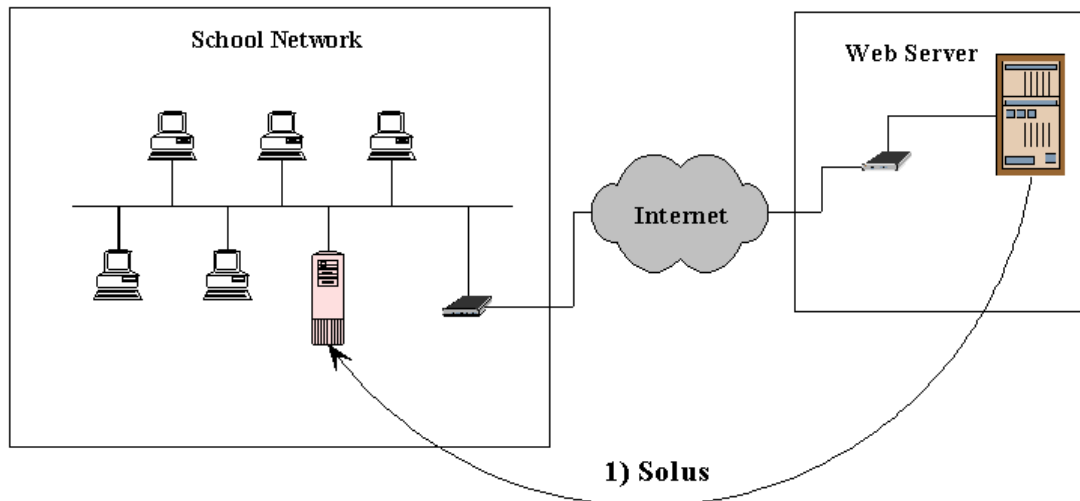
- The school has applied the upgrade successfully.
- The school has applied the upgrade and the upgrade failed.

## Using the SOLUS 2 Wizard to Upgrade your SIMS System

For instructions on how to run the SOLUS 2 wizard, which is used to upgrade your SIMS system to the latest release of the software, please refer to the *SIMS SOLUS 2 User Guide*. This document is available from the SOLUS web site, [www.capitasolus.co.uk](http://www.capitasolus.co.uk), by clicking the **SOLUS 2 Documentation** link and clicking the document title.

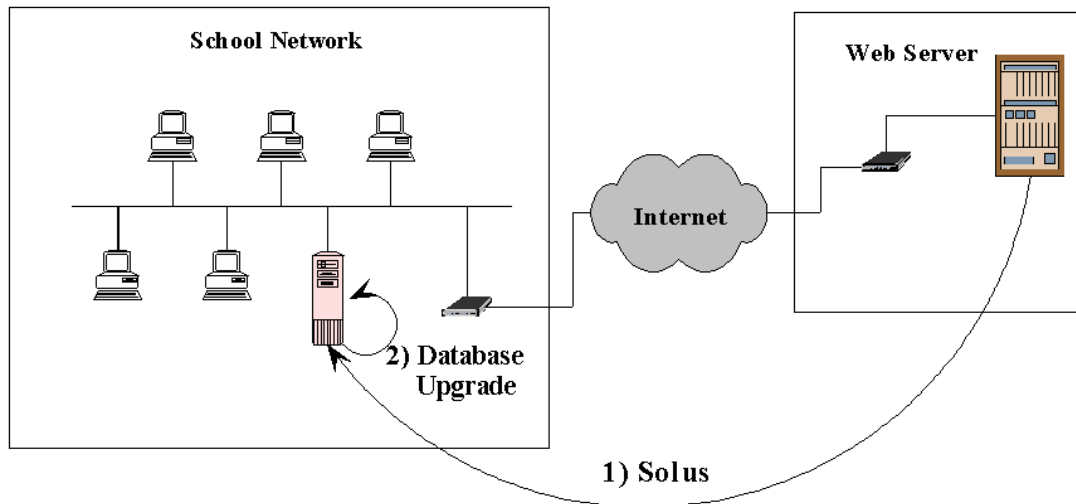
For information on how the download of updates from the SOLUS website can be scheduled to run automatically, please refer to *Using Scheduled Tasks to Automate the SOLUS Download Process* on page 24.

## What Happens During a Download from the SOLUS Website?



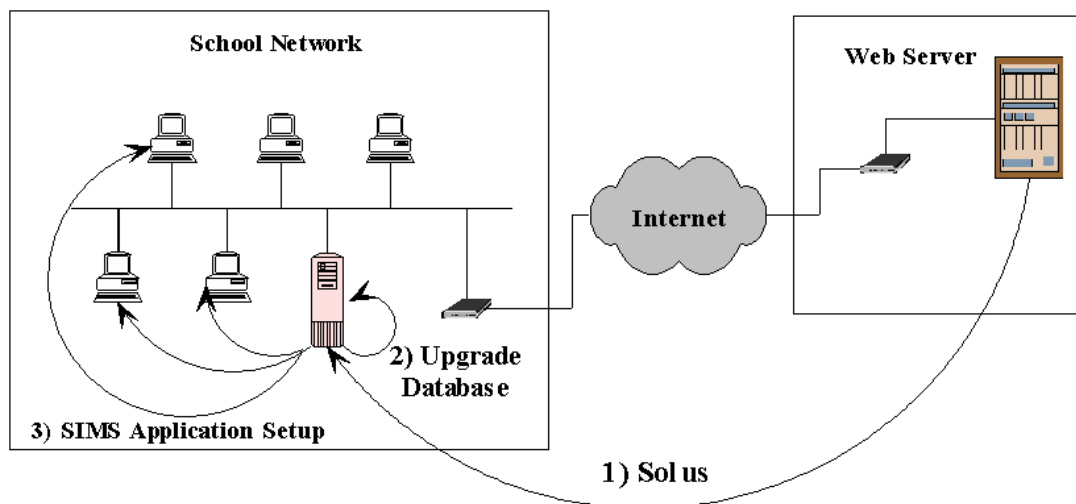
When an update to the SIMS system is made available, schools can download it and apply the upgrade using the SOLUS 2 wizard. This is achieved by selecting **Tools | Check for Updates** in SIMS .net, or by selecting **Start | All Programs | SIMS Applications | Update and Run SOLUS** from the Taskbar.

After entering your login details, the download process begins. The files are saved in the `\SIMS\Setups` folder.



Once the download has completed successfully, the database upgrade is carried out automatically on the SIMS SQL server.

Once the database has been upgraded, the SIMS SQL Application setup is run. This setup routine copies shared SIMS executable files to the \SIMS folder on the SIMS server and also updates additional SIMS SQL applications (i.e. dbUpgrade, dbAttach, etc.) that are stored in the MSSQL\Binn folder on the SIMS SQL server.



The final part of the upgrade process involves updating each SIMS workstation in the school. It is this part of the upgrade process that is covered in depth in this document.

Files that are upgraded as part of a SIMS workstation setup include the SIMS .net executable files, user handbooks and help files.

For more information on the purpose of each of the setup routines that are downloaded as part of a SIMS release, please refer to *What is the Purpose of the Files in the \SIMS\Setups Folder?* on page 15.

## Using Scheduled Tasks to Automate the SOLUS Download Process

For a number of reasons, it might not be convenient to download the new release of SIMS .net from the SOLUS server and to run the SIMS database upgrade during the working day. For a database upgrade to work successfully, all users must be logged out of SIMS and this might not be possible during the day. You might also find that the download of an update from the SOLUS server takes too long during the day, as network traffic is usually at its busiest.

To cater for scenarios such as these, it is possible to automate the SOLUS download process and schedule it to run at a more convenient time, i.e. overnight using a simple batch file which is set to run at a time defined in Windows Scheduler.

A batch file can be created in any plain text editor, such as Windows Notepad. The following is a **sample** batch file, which when run, will download a release from the SOLUS server and perform a server and database upgrade:

```
SOLUS♦/user=solus♦/password=abcd♦/auto=downloadandupgrade♦  
/agreetoterms♦/forceupgrade♦/sauser=sa♦/sapassword=password
```

where:

**SOLUS** is the name of the executable file which the batch file is calling (SOLUS.EXE).

**/user=solus** is the name of the SIMS user carrying out the upgrade.

**/password=abcd** is the password of the SIMS user carrying out the upgrade.

**/auto=downloadandupgrade** informs the SOLUS server that you want to both download the new release and upgrade the server and database.

**/agreetoterms** automates the acceptance of the terms and conditions.

**/forceupgrade** forces any users out of SIMS who are still logged in.

**/sauser=sa** is the login name of the SQL system administrator.

**/sapassword=password** indicates that the SQL system administrator's password is **password**.

---

**IMPORTANT NOTE:** In the previous example, all spaces are indicated by the ♦ symbol.

---

Once the file has been saved with the file extension .BAT (e.g. SOLUSDownload.BAT), it should be placed in the same location as the SOLUS.EXE file (e.g. \Program Files\SIMS\SIMS .net) or in a location where you have access to the SOLUS web server.

For instructions on how to set up the batch file, and a comprehensive list of parameters that can be added to the batch file, please refer to the *Automating SIMS SOLUS 2*. This document is available from the SOLUS web site, [www.capitasolus.co.uk](http://www.capitasolus.co.uk), by clicking the **SOLUS 2 Documentation** link and clicking the document title.

A scheduled task can now be defined in order to run the batch file at the required time.

1. On the SOLUS server, select **Start | All Programs | Accessories | System Tools | Scheduled Tasks** from the Taskbar to open the **Scheduled Tasks** window.
2. Double click the **Add Scheduled Task** icon to open the **Scheduled Task** wizard.



3. Click the **Next** button to continue.



4. Search for the batch file you created by clicking the **Browse** button to open the **Select Program to Schedule** dialog.

5. Navigate to the location of the required batch file, highlight the file and click the **Open** button.

*NOTE: This should be the \Program Files\SIMS\SIMS .net folder on the workstation and should be the same location as the SOLUS.EXE file.*



6. Edit the task name, if required. This task name defaults to the name of the program selected.
7. In the **Perform this task** section, select the radio button relating to how often you wish to run this task.
8. Click the **Next** button to continue.



This page will vary depending on the radio button selected in the **Perform this task** section on the previous page. The previous example shows the page that will be displayed if the **Monthly** radio button is selected.

9. Complete the details as required and click the **Next** button to continue.



10. The **Enter the user name** field will be populated automatically from your current Windows login name, but this can be edited if required.
11. Enter your Windows password in the **Enter the password** field and confirm it in the **Confirm password** field.
12. Click the **Next** button to continue.



A message will be displayed confirming the set up of the scheduled task, together with details of when it will be run. If any of these details are incorrect, click the **Back** button and amend the details on the previous pages.

13. View or modify the advanced properties, if required, by selecting the **Open advanced properties for this task when I click Finish** check box.
14. Click the **Finish** button to complete the process.



## Glossary

### **CONNECT.INI**

The `CONNECT.INI` file tells SIMS .net which SIMS SQL database to use with the application. It is *essential* that the correct database is referenced in the `CONNECT.INI` file, to ensure that the current, live database is being used to store accurate and up-to-date data.

### **SIMS .net Folder**

See *SIMS Workstation Application Folder*.

### **SIMS Document Server Application Folder**

This is where the SIMS .net document server programs are installed. This is not a file share and does not need to be directly accessible by the workstation applications. Typically this is located at `C:\Programs Files\SIMS\SIMS .net Document Server`.

### **SIMS Document Server Document Folder**

This folder is where the document server keeps all its compressed encrypted documents; it does not need to be visible from the workstation. This is typically located at `C:\DocStorage`.

### **SIMS Fileserver Application Folder**

This is where the historical Common Platform 2 .EXE files and the dBase data files are stored, together with the centrally stored copies of user handbooks, etc. Typically this is accessed through a file share on a network set to `S:\SIMS`. On standalone machines, this folder is typically `C:\SIMS`. Full access must be granted to this folder for those SIMS users who still require access to the following areas:

- FMS
- DENI
- System Manager for DOS
- Launcher
- Personnel 5
- archived STAR students
- Web Integrator for Parents Gateway.

### **SIMS Folder**

See *SIMS Fileserver Application Folder*.

### **SIMS Setups Folder**

This folder contains a copy of all the Setup programs. Typically it will be located at `S:\SIMS\Setups`.

**SIMS SQL Server Application Folder**

This is where back end program files are located for SIMS .net. These are run through timed jobs by SQL Server Agent and are located in the same program folder as SQL Server itself. Typically this is C:\Program Files\Microsoft SQL Server\MSSQL\Binn. This is one of the few folders that can be definitely 'seen' by SQL Server Agent and is therefore where SIMS .net puts its program files.

**SIMS Workstation Application Folder**

This is located on the workstation, typically at C:\Program Files\SIMS\SIMS .net. This is where all workstation binaries and initialisation files are kept. A user operating SIMS will need read/write access to this folder. Common Platform 3 applications (i.e. System Manager, Attendance, etc.) will have their COM .DLL and .TLB files stored in this folder, together with their .EXE files.

**SIMS.INI**

SIMS.INI contains important information relating to the SIMS system, such as the location of the local SIMS .net folder, the location of the \SIMS\Setups folder on the SIMS server the location of the SIMS folder on the SIMS server. This file can be found in the \WINDOWS folder on the workstation.

**SIMSLOAD.INI**

SIMSLOAD.INI stores information on which setup routines have been run on the workstation. The most recently run versions of SIMSApplicationSetup, SIMSManualSetup and SIMSAmparkSetup are recorded. This file can be found in the \Program Files\SIMS\SIMS .net folder on the workstation.

A SIMS .net desktop shortcut can be amended so that it can trigger an automatic workstation setup on opening SIMS .net, by adding a reference to SIMSLOAD. When run, the shortcut forces the system to look at SIMSLOAD.INI and compares its contents with the contents of the SIMS\Setups folder on the SIMS server. If the versions of the upgrade files in \SIMS\Setups are later than those stored in SIMSLOAD.INI, this indicates that upgrades have not yet been run on the workstation, so the workstation upgrade is performed automatically. If the versions are the same in \SIMS\Setups and SIMSLOAD.INI, the workstation is up-to-date and therefore no workstation upgrade is required.

**SIMSSQLLOAD.INI**

SIMSSQLLOAD.INI stores information on the latest SIMSSQLApplicationSetup routine that was run on the SQL Server. This file can be found in the \Program Files\Microsoft SQL Server\MSSQL\Binn folder on the SIMS SQL server.

**Windows Folder**

This is where SIMS.INI is maintained and a user operating SIMS will need read/write access to this file. This folder is defined by Windows.

## Index

### A

- access rights
  - granted by SIMSPerm.BAT ..... 5
  - granted to files ..... 6
  - granted to folders..... 5
  - granted to registry..... 5
  - running workstation upgrades ..... 1
  - setting the system, for users..... 4
- AD provisioning service setup files
  - SIMS setups folder ..... 16
- adding
  - SIMS .net shortcuts..... 14
- AMPARK setup files
  - SIMS setups folder ..... 17
- application setup files
  - SIMS setups folder ..... 18
- automating
  - SIMS upgrades ..... 9
  - SOLUS downloads ..... 24

### C

- checking
  - CONNECT.INI ..... 4
  - drive mappings..... 3
  - SIMS .net shortcuts..... 4
  - SIMS.INI file..... 3
- command line scheduler
  - using to automate SIMS upgrades .... 9
- CONNECT.INI
  - checking ..... 4
  - master ..... 13
  - redirect ..... 13
  - troubleshooting..... 11
- consequences
  - not running SIMSPerm.bat ..... 8
- considerations
  - before upgrading a SIMS workstation 1
  - CONNECT.INI ..... 4
  - drive mappings..... 3
  - SIMS .net shortcuts..... 4
  - SIMS.INI file..... 3
- creating
  - SIMS .net shortcuts..... 14

### D

- database upgrade files
  - SIMS setups folder ..... 15
- disconnecting a network drive ..... 11
- document server setup files
  - SIMS setups folder ..... 18
- downloading upgrade files
  - overview..... 21
- downloads
  - automating ..... 24
- drive mappings
  - checking..... 3
- drives
  - disconnecting network ..... 11
  - mapping network..... 10

### F

- files
  - access rights ..... 6
  - overview of downloading from SOLUS  
..... 21
- folders
  - access rights ..... 5

### G

- granting
  - access rights to files ..... 6
  - access rights to folders..... 5
  - access rights to the registry ..... 5
  - access rights using SIMSPerm.BAT.. 5

### I

- infrastructure setup files
  - SIMS setups folder ..... 18
- installer files
  - SIMS setups folder ..... 19

### L

- launching
  - upgrades, using SIMS .net shortcuts. 4

### M

- manual setup files

- SIMS setups folder ..... 19
- mapping a network drive ..... 10
- master
  - CONNECT.INI file ..... 13
- modules
  - SIMS .net, upgrading ..... 8
- MS SQL desktop engine setup files
  - SIMS setups folder ..... 16
- N**
- network drives
  - disconnecting ..... 11
  - mapping ..... 10
- Nova-T satellite setup files
  - SIMS setups folder ..... 17
- O**
- overviews
  - SOLUS ..... 21
- P**
- permissions
  - running workstation upgrades ..... 1
- R**
- redirect
  - CONNECT.INI file ..... 13
- registry
  - access rights ..... 5
- running
  - SIMSPerm.BAT ..... 7
- S**
- scheduled tasks
  - using to automate SOLUS downloads ..... 24
- setting
  - system access rights for users ..... 4
- shortcuts
  - checking ..... 4
  - creating ..... 14
- SIMS
  - upgrading using SOLUS ..... 22
- SIMS .net
  - upgrading modules ..... 8
- SIMS .net shortcuts
  - creating ..... 14
- SIMS setups folder
  - AD provisioning service setup files ..... 16
  - AMPARK setup files ..... 17
  - application setup files ..... 18
  - database upgrade files ..... 15
  - document server setup files ..... 18
  - infrastructure setup files ..... 18
  - installer files ..... 19
  - introduction ..... 15
  - manual setup files ..... 19
  - MS SQL desktop engine setup files ..... 16
  - Nova-T satellite setup files ..... 17
  - SQL application setup files ..... 19
- SIMS upgrades
  - automating ..... 9
- SIMS workstation upgrade
  - troubleshooting ..... 9
- SIMS.INI
  - troubleshooting ..... 9
- SIMS.INI file ..... 3
- SIMSPerm.BAT
  - access rights granted ..... 5
  - access rights to files ..... 6
  - access rights to folders ..... 5
  - access rights to the registry ..... 5
  - consequences of not running ..... 8
  - introduction ..... 4
  - running ..... 7
  - what is it? ..... 5
  - when it should be run ..... 6
  - who can run it? ..... 6
- SOLUS
  - automating downloads ..... 24
  - overview ..... 21
  - using the wizard to upgrade ..... 22
- SQL application setup files
  - SIMS setups folder ..... 19
- system access rights
  - setting for users ..... 4
- T**
- troubleshooting
  - CONNECT.INI file ..... 11
  - SIMS workstation upgrade ..... 9
  - SIMS.INI file ..... 9

**U**

- understanding
  - master CONNECT.INI file ..... 13
  - redirect CONNECT.INI file ..... 13
- upgrade files
  - overview of downloading ..... 21
- upgrades
  - launching, using shortcuts..... 4
  - SIMS, automating ..... 9
  - using SOLUS ..... 22
- upgrading
  - access rights required ..... 1
  - SIMS .net modules..... 8
- upgrading workstations
  - considerations ..... 1
- users
  - setting system access rights..... 4
- using
  - SOLUS 2 wizard to upgrade ..... 22
- using command line scheduler to automate SIMS upgrades ..... 9

- using scheduled tasks to automate SOLUS downloads ..... 24

**W**

- what is SIMSPerm.BAT? ..... 5
- what is the SIMS setups folder? ..... 15
- what will happen if i do not run SIMSPerm.BAT ..... 8
- when should SIMSPerm.BAT be run? .... 6
- who can run SIMSPerm.BAT? ..... 6
- windows explorer
  - disconnecting network drives..... 11
  - mapping network drives..... 10
- wizards
  - using SOLUS 2 to upgrade ..... 22
- workstation upgrade
  - troubleshooting ..... 9
- workstations
  - considerations before upgrading ..... 1
  - permissions required for workstation upgrades ..... 1