

Technical Guide – SIMS Installation Tools and Procedures 2004

Module	Standard Architecture
Author	Phillip Hamlyn Chief Architect
Revision Number	4.0
Last Edited	11 th February 2004

Technical Guide – SIMS Installation Tools and Procedures 2004.....	1
Purpose.....	2
Background.....	3
SIMS Installation Environment	4
The SIMS Workstation Application Directory	4
The Windows Directory.....	4
The SIMS Fileserver Application Directory.....	4
The SIMS SQL Server Application Directory	4
The SIMS Setups Directory	4
The SIMS Document Server Application Directory	4
The SIMS Document Server Document Directory.....	4
Installation Programs	5
SIMSInfrastructureSetup.EXE	5
SIMSApplicationSetup.EXE.....	5
SIMSSQLApplicationSetup.EXE	6
SIMSDocumentServerSetup.EXE	6
SIMSManualSetup.EXE	6
SIMSInstaller.EXE	6
DBUpgrade.EXE.....	7
Associated Programs and Auto-Updating	8
Initialisation Files	8
SIMS.INI	8
Connect.INI.....	8
SIMSLoad.INI	8
SIMSSQLLoad.INI.....	9
Automatic Updating.....	9
Workstation Upgrading	9
SQL Server Programs Upgrading	10
Digital Signatures.....	10

Purpose

This document is issued by the CES Architecture team to act as a guide to the purpose, behaviour and technology of the SIMS installation tools issued as part of the Spring CD 2004. This document does not cover FMS, NOVA or Common Platform 2 installation technologies which are unchanged for Spring CD 2004.

This document is intended to be made available to CES customers.

Background

The installation procedures of the SIMS suite cope with a large range of different technology platforms and designs of application. In 2004 the tools have been redesigned to

Make them more user-configurable for complex environments such as terminal server, and centralised SQL Server hosts.

Make them more predictable in their security requirements.

Make them separable so customers with pre-configured network environments could alter the way in which they deploy SIMS.

In order to achieve this several major changes will occur after installing the February 2004 release of SIMS.

1. All workstation components, programs, files etc. will be installed into a single directory instead of the multiple directories as previously.
2. Some application components will be installed on the SQL Server PC and will interoperate with the workstation through the SQL Server Agent utility.
3. No files will be stored on the workstation in protected (i.e. Common or Windows System) directories.
4. Every separate SIMS installation tool can be run independently and do not automatically call one another in order to aid users in complex environments not envisaged by the SIMS design team (serviced networks, server clusters etc.)

SIMS Installation Environment

During the SIMS installation there are seven significant locations. These may be on separate physical computers. This glossary of terms will be used within this document to identify the specific directory locations.

The SIMS Workstation Application Directory

This is located on the workstation, typically "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 directory. Common Platform 3 (i.e. Attendance, Profiles, System Manager, etc.) applications will have their COM DLL and TLB files stored in this directory.

The Windows Directory.

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

The SIMS Fileserver Application Directory.

This is where the Common Platform 2 (i.e. Launcher, Data Manager, etc.) program files, Common Platform 3 EXE files, and the dBase data files are kept. A user operating SIMS will need full rights to this directory. Typically this is accessed through a file share on a network set to "S:\SIMS". On standalone machines this directory is typically "C:\SIMS".

The SIMS SQL Server Application Directory

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 directory as SQL Server itself. Typically this is "c:\program files\microsoft sql server\mssql\binn". This is one of the few directories that can be definitely "seen" by SQL Server Agent and is therefore where SIMS.net puts its program files.

The SIMS Setups Directory

This is a new directory for February 2004 and contains a copy of all the Setup programs listed below. Typically it will be "S:\SIMS\SETUPS".

The SIMS Document Server Application Directory

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 "c:\program files\sims\sims .net document server".

The SIMS Document Server Document Directory

This directory is where the document server keeps all its compressed encrypted documents; it does not need to be visible from the workstation. This is typically "c:\docstorage".

Installation Programs

There are seven separate programs associated with installation ;

SIMSInfrastructureSetup.EXE

This is written in InstallShield 6 and contains a compressed single file setup containing all the technologies not authored by CES that SIMS CP2, CP3 and SIMS.net requires. This installs the following and requires local Administrator access to the PC

- Visual Basic 6 Runtime Engine
- Borland Database Engine 5.11
- Windows 98 Year 2000 Patch
- Microsoft Data Access Components 2.7 ("RTM Refresh") (includes Microsoft XML Parser V3)
- Microsoft Jet Database Engine 4.00
- Microsoft Installer Engine V2
- Microsoft XML Parser V2
- Microsoft XML Parser V4
- Crystal Reports Runtime Engine V5
- Windows Scripting Host 5.6
- Windows Management Instrumentation 1.5
- Microsoft .net Common Language Runtime V1.1
- DynaZip 5

Various OCX user controls and DLLs installed into Windows System directory.

This infrastructure is required by both the SIMSApplicationSetup.EXE and SIMSSQLApplicationSetup.EXE and is the same in either the workstation or SQL Server environments. In the Spring CD this program will additionally uninstall the SIMS Workstation Setup feature and Lesson Monitor application and database.. This is to remove the footprint of the previous versions of SIMS from the workstation so that they can be relocated to a single unambiguous and low security directory. For the Spring CD only access rights to carry out the above will be required. This approach gives the SIMS support teams a single set of infrastructure to be deployed on a SIMS PC. We do not intend to change this setup program often, aiming for a maximum of a single change per year.

SIMSApplicationSetup.EXE

This is written in Wise Installation System and deploys all the SIMS workstation applications and online help files to the SIMS Workstation Application Directory and common platform 3 EXE files to the SIMS Fileserver Application Directory. This deploys all SIMS applications in a single step and can take a command line parameter to indicate which directory to install the programs to. If this parameter is omitted a user interface will ask the user to nominate the destination directory. This also sets up the systems for All Users use and creates start menu icons appropriately.

This setup program requires the following rights

1. Full rights to the SIMS Workstation Application Directory.
2. Read/Write access to the SIMS.INI file in the Windows directory.
3. Read/Write access to the following registry keys

- HKEY_LOCAL_MACHINE\SOFTWARE\Classes
- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID
- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface
- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib

This setup program may be run more than once on the workstation (see Auto-Updating below) so it is recommended that users of the PC who also use SIMS are granted these rights permanently if you are running a secure network environment.

This setup program will be re-issued at various intervals for fault-fixes and new features.

This installer will write it's file version number into the file SIMSLoad.INI (see auto-updating below) on successful completion.

SIMSSQLApplicationSetup.EXE

This is written in Wise Installation System and deploys all the SIMS.net programs that operate on the SQL Server. These are a mixture of extended stored procedures (DLL files) and Business to Business module components (.net DLL and EXE files). The installer requires the same security permissions as SIMSApplicationSetup.EXE and will be re-issued at various intervals for fault-fixes and new features.

This installer can take a command line parameter to indicate which directory to install the programs to; otherwise it will use a UI to ask the user for the location to install the applications to. This must be the same as the location of the SQL Server programs in order for SQL Server Agent to locate the SIMS programs relative to its locations.

This installer will write it's file version number into the file SIMSSQLLoad.INI (see auto-updating below) on successful completion.

SIMSDocumentServerSetup.EXE

This is written in InstallShield 6 and installs the SIMS.net document server programs into a Windows Service into the SIMS Document Server Application Directory. The user is prompted for this program location as part of the setup. The SIMSDocumentServerSetup.EXE must be run with local System Administrator security because it sets up and maintains a Windows service.

It is unlikely that the SIMSDocumentServerSetup.EXE will be changed as frequently as the SIMSApplicationSetup.EXE.

SIMSManualSetup.EXE

This is written in Wise Installation System and deploys the optional manuals and handbooks to the SIMS Workstation Application Directory so they can be called from the online help system and from within the application. The size of this documentation and the fact that many customers may prefer to print it for reference has led to it being split into a separate installer program.

SIMSInstaller.EXE

This is written in Delphi 5 and is a user-friendly wrapper which automates the choice of which of the above installations to run. It asks the user four questions

1. Is this machine to be used as a SIMS SQL Server ?
 - a. Which SQL Server do you want SIMS to use ?

This will run SIMSInfrastructureSetup.EXE and SIMSSQLApplicationSetup.EXE

2. Is this machine to be used as a SIMS Document Server ?
 - a. Where do you want to install the SIMS Document Server ?
 - b. Where do you want to store your SIMS Documents ?
 - c. Which HTTP Port to use ?

This will run SIMSInfrastructureSetup.EXE and SIMSDocumentServerSetup.EXE

3. Is this machine to be used as a SIMS Workstation ?

- a. Where do you want to install your SIMS Workstation Programs ?

This will run SIMSInfrastructureSetup.EXE and SIMSApplicationSetup.EXE

- b. Do you want to install the SIMS manuals in addition to the online help files ?

This will run SIMSManualSetup.EXE

The user can select any combination of uses for the machine in question and the SIMSInstaller will coordinate the execution of the various setup packages.

DBUpgrade.EXE

This program carries out database upgrade installation. This is installed into the SIMS Workstation Application Directory and the SIMS SQL Application Directory by the associated installers and is designed to check the version number of the database, check the SIMS Setups Directory for suitably named files and apply the scripts it finds. Technically this doesn't require to be run on the SQL Server PC (unlike the previous dbMove utility) but it does require the .net framework to be installed and honours the settings in SIMS.INI and Connect.INI but does require a SIMS user name and password login (all of which can be supplied on the command line). A separate document "Technical Guide - The SIMS.net Database Upgrade Tool" is available that discusses in detail its operation.

It is run in three different circumstances ;

1. If you run SIMSSQLApplicationSetup.EXE from SIMSInstaller.EXE, the SIMSInstaller program will automatically run DBUpgrade afterward.
2. If you run the SOLUS utility (documented elsewhere) and it causes a database upgrade file to be downloaded it will run run DBUpgrade afterward.
3. In subsequent CDs; if the user runs CD installation wrapper that deposits the new setup programs to the SIMS Setups Directory from a SQL Server or from a Workstation (i.e. there is a SIMSDotNetDirectory setting present in SIMS.INI - see "Initialisation Files" below) then DBUpgrade will be run afterward. If the CD installation wrapper is not being run from a SIMS enabled machine (i.e. some kind of administration server) then DBUpgrade cannot be run automatically and the user will need to run the tool manually from their SQL Server or Workstation PCs.

Associated Programs and Auto-Updating

Initialisation Files

SIMS.INI

The SIMS.INI file in the Windows directory has three critical entries in it that allow the SIMS suite to seamlessly install. These are

[Setup]

SIMSDotNetDirectory=c:\program files\sims\sims .net

SIMSDirectory=s:\sims

SIMSSetupDirectory=\\simsFServer\simsroot\sims\setups

(This is held as a UNC path to allow SIMSSQLLoad executing in the system context to perform version checking against available installation packages as user specific drive mappings will not be available)

SIMSDotNetDirectory

This records the location of the SIMS Workstation Application Directory and is used to prevent the user having to nominate the location of SIMS when running a new version of the SIMSApplicationSetup.EXE. This setting is stored by the SIMSApplicationSetup.EXE program when its successfully installed. This setting is also used by third party tools and programs that need to call into the SIMS.net code to gain access to the SIMS database.

This setting is not used to locate the SIMS SQL Server applications which are always found with reference to the SQL Server program files directory.

SIMSDirectory

This records the location of the SIMS Fileserver Application Directory and is used to help SIMS.net to find the Common Platform 3 EXE files (Attendance, Profiles, System Manager etc.) . It is used by the SIMSApplicationSetup.EXE installer to prevent the user having to nominate the location of SIMS on the fileserver when installing a new version of the SIMS programs.

SIMSSetupDirectory

This is used to allow SIMSLoad and SIMSSQLLoad (see Automatic Updating below) to find the latest versions of the SIMS installer files and compare them against the last ones they ran.

Connect.INI

In addition to these the Connect.INI file in the SIMS Workstation Application Directory or the SIMS SQL Application Directory containing database and server connection information is used by DBUpgrade.

SIMSLoad.INI

The SIMSLoad.INI file in the SIMS Workstation Application is used to record the version number of the last SIMSApplicationSetup.exe package to run. This is written by SIMSApplicationSetup.exe.

[LastChecked]

SIMSApplicationSetup=1.0.0.0

SIMSSQLLoad.INI

The SIMSSQLLoad.INI file in the SQL Server programs directory is used to record the version number of the last SIMSSQLApplicationSetup.exe package to run. This is written by SIMSSQLApplicationSetup.exe.

```
[LastChecked]  
SIMSSQLApplicationSetup=1.0.0.0
```

Automatic Updating

Workstation Upgrading

In the past the program "Launcher" would check for new versions of the setup programs through an INI file setting. With Launcher being retired in the future and SIMS.net being used for all external application starting this method of workstation upgrading has been changed. One of the problems with the previous system is that all the files required to set up a brand new workstation were wrapped into a single "Workstation Setup" installation which contained the workstation infrastructure. Running this installation therefore required local administrator access to the machine.

It is CES policy to now change the SIMS Infrastructure delivered through SIMSInfrastructureSetup.Exe as little as possible in order to cut down the number of workstation visits required by LAN administrators in a secure environment. This is part of the reason that the SIMS programs are now delivered through a different setup program to the infrastructure.

Users of the SIMS suite will start one of two programs on the workstation, either SIMS.net or Lesson Monitor. To prevent both of these programs from incorporating version checking information and in order to allow systems administrators to intervene in this automatic upgrading (should they wish so), this is now done in a completely separate utility called SIMSLoad.

SIMSLoad.EXE is a simple .net application installed by the SIMSApplicationSetup.EXE installer. It checks the file version number in the SIMSLoad.INI file (written by the SIMSApplicationSetup installer) in the SIMS Workstation Application Directory and cross checks it with the file version number of the SIMSApplicationSetup.EXE file held in the SIMS Setups Directory. If the file present in the SIMS Setup Directory is higher than that recorded in SIMSLoad.INI then it will be executed. SIMSLoad will wait for it to complete and it will then run the passed program file.

SIMSLoad takes a single parameter, the name of the application to run after it has completed its checks. It is used in the Start Menu to prefix the SIMS applications ; for example

Start->Programs->SIMS Applications->SIMS .net shortcut actually runs the following command line
"C:\program files\sims\sims .net\simsload" "c:\program files\sims\sims .net\pulsar.exe".

SIMSLoad is passed the name of the program the user wants to run and carries out it's version checking before the program is executed. In order to be as transparent as possible SIMSLoad passes the SIMSApplicationSetup.EXE it's own application location to prevent it needing to ask the user where to install SIMS.net to.

If the SIMSApplicationSetup.EXE file cannot be located, or there is no entry in SIMS.INI for the SIMS Setups Directory then it skips the version checking stage. If there is no parameter sent to SIMSLoad it will carry out the version checking and then do nothing.

This tool makes it simple for a LAN administrator to separate the automatic upgrade of the workstation applications from the running of those applications. A LAN administrator can remove the SIMSLoad part of the Start Menu shortcut and execute SIMS.net or Lesson Monitor directly, and alternatively run SIMSLoad as part of their user's network login script.

SQL Server Programs Upgrading

The SQL Server database is upgraded by DBUpgrade (see above) and is separate from the upgrading of the SIMS programs run on the SQL Server.

The SQL Server SIMS programs are upgraded in a very similar way to the workstation using a similar tool called SIMSSQLLoad. Because all SQL Server programs are run from within SQL Server Agent there are no Start Menu shortcuts to see. When a SIMS job is run from within SQL Server Agent it runs the following command line;

```
"C:\program files\....\binn\SIMSSQLLoad.EXE" "c:\program files\...\binn\b2b.exe"
```

SIMSSQLLoad carries out the same task as SIMSLoad but checks "SIMSSQLLoad.INI" and "SIMSSQLApplicationSetup.EXE" instead of SIMSLoad.INI and SIMSApplicationSetup.EXE.

Again the LAN administrator can intervene in this procedure but cannot edit the command line directly. If a user wants to prevent the version checking of the SQL Server applications they need to edit the SIMS.INI file and remove the SIMSSetupDirectory entry.

Digital Signatures

In any automatic system upgrade procedure there is a risk that the tools that propagate the new programs could be subverted to distribute software not issued by Capita. This threat is made more significant where SOLUS is used to supply a school with software that is then automatically distributed to all the SIMS workstations within the school.

In order to prevent this form of illegal attack, the tools used in the automatic updating of SIMS will only operate on files that have an associated digital signature file. The specific tools are ;

SOLUS - the internet upgrade wizard

SIMSLoad - the workstation upgrade tool.

SIMSSQLLoad - the SQL Server upgrade tool.

DBUpgrade - the database upgrade tool.

SIMSConfiguration - the database configuration tool.

All the files that these tools use are provided by Capita with a digital signature file. These can be recognised as having the same name as their subject followed by a ".signature" file name extension. For instance the database upgrade file "3.61.014 to 3.61.015 upgrade.zip" is supplied with the digital signature file "3.61.014 to 3.61.015 upgrade.zip.signature". If the tools above attempt to open or execute files in the SIMS Setups directory that are missing their signature file, or if the signature contained in those files are invalid, they will give an appropriate error message and stop their operation.

The content of the digital signature files are not readable and cannot be predicted. They are created by the SIMS distribution team at the point a release is made public (either on a CD or on the SOLUS web site) using industry standard digital signature algorithms based on public key cryptography and hashing of the binary image of the file to be distributed. This technical algorithm means that

1. Files with a signature that our tools validate cannot have been produced by any other Company.
2. Files with a signature that our tools validate cannot have been altered since they were issued.

The software distributed by Capita cannot be decompiled to reveal any useful information about the creation of the digital signature and the private key used to sign the SIMS files is not provided to third parties (or even other teams within Capita).