Assimilator 2.0 from assimilator.com

Introduction

Purpose

Assimilator is designed for situations where you wish to make a large number of Macintoshes look virtually identical. Typically this is in an educational laboratory situation -- this document is aimed at laboratory administrators -- but Assimilator has other uses, such as setting up newly purchased Macs, testing software or setting up demo machines.

This document is rather long but it pays to read the entire thing through at least once. It is full of helpful hints and tips about setting up Assimilator and your Macintosh laboratory. It is divided into two main sections, the first introduces the major functions of Assimilator and the second introduces some of the more complex ways you can customise how Assimilator operates (mostly features introduced in version 2.0 of the software).

We have endeavoured to make Assimilator as easy to use as possible, but the task it performs is surprisingly tricky. Once again we emphasise that you should read the documentation in full, at least once. This will save you from pain and misery in the future.

How it Works

When Assimilator runs on a laboratory Macintosh it will mount an AppleShare file server and then make the hard disk look virtually identical to a pre-specified source folder on the server. It does this by comparing each file in the source folder to the corresponding file on the destination hard disk. If the file on the destination is missing, or has been changed, then it is thrown away (into the Trash) and a new copy of the file is copied down from the source. Any extraneous items on the destination are also thrown away. Finally aliases, icon positions, folder views, Finder flags and so forth are all set to match those in the source folder. This entire process is known as **assimilation** .

Restrictions

Assimilator makes no attempt to secure its files from malicious users. If you have malicious users, then you will need to take additional steps to protect your system, including installing additional software like LabMaster and making your System Folder invisible and so forth. Note that Mac OS 8.x will not run Startup Items or Extensions if the system folder or Startup Items/Extensions folder is invisible.

Assimilator requires System 7 and a hard disk.

Using Assimilator

What Comes in the Package

The following files are including in the Assimilator distribution:

README!
A warning about the dangers of running Assimilator without reading the documentation.

Assimilator Admin
The main program that controls Assimilator.

Documentation
This document.

Register
A program that you should use when you decide to register Assimilator.

Programs
A list of our other public domain, freeware, shareware and commercial products.

You'll notice that there is no "Assimilator" program contained in the distribution. This is because you use the Assimilator Admin program to generate Assimilators that are customised for your environment.

Creating a Customised Assimilator

When you run Assimilator Admin, it opens a (new) window for you to enter the necessary information for creating your own customised Assimilator. This window contains the following five areas:

Source
This panel lets you specify the source folder for the Assimilator. You must fill out this before you can create an Assimilator that performs any useful action. Creating a suitable source folder is discussed in the next section.

Subfolder

This lets you specify a subfolder to be Assimilated, rather than the entire source folder. See the Customization Toolkit: Subfolders section for more information.

Kind
The controls in this panel specify what type of file this Assimilator will be saved as. Understanding this is critical to understanding how Assimilator works and it is discussed later in this section.

Miscellaneous
This area lets you set various sundry preferences. It is fully described in a later section.

Label Actions
This area lets you associate specific action with specific Finder labels. These actions are described fully in a later section but normally the default actions are OK.

Edit Database
Each Assimilator also holds a database that it consults when it wants to set machine specific parameters. The database is discussed in a later section.

Don't Restart
Normally after assimilating a machine Assimilator will restart the machine. This is because Assimilator often changes portions of the System Folder. However, if you know Assimilator will not be affecting the System Folder you may which to force Assimilator not to restart by using the Don't Restart checkbox.

The key to understanding Assimilator is the **Kind panel** . Assimilator Admin works on documents -- that is, customised Assimilators -- which are themselves either documents, applications or extensions. When you save a customised Assimilator from within Assimilator Admin it creates the appropriate kind of file, depending on the settings in the Kind panel.

If you save your customised Assimilator as an **application** then it will start Assimilating whenever it's launched. You can put it in the Startup Items folder (to run at system startup), the Shutdown Items folder (to run at system shutdown) or in a convenient place, so that users can run it manually. If you run Assimilator as a Shutdown Item you also need to put the Assimilator Helper extension in the Extensions folder (the extension tells Assimilator if the machine is being shutdown or restarted).

One problem with putting Assimilator in the Startup or Shutdown Items folders is that it won't run if it is invisible. [The Finder refuses to launch any invisible items.] To get around this you can save Assimilator as a **system extension** . Assimilator loads a small nub at startup time (like any other system extension) and waits for startup time to finish. It then automatically launches itself, which then assimilates the system.

Warning : Mac OS 8.x will not run Extensions if the System Folder or Extensions folder is invisible.

Note: Assimilator cannot run at Shutdown time if it is a system extension.

Finally you can save the Assimilator as a **document** , which contains just the configuration information. This is a useful way of storing and transporting a configuration without any danger of it accidentally being executed. Also, at some points Assimilator Admin will only let you save a configuration as a document because the configuration does not contain enough information to be a valid Assimilator.

Creating a Source Folder

Once you have created your new untitled Assimilator, you need to start filling out some of the fields. The first step is to create a **source folder** . This is a folder on a file server that acts as the master folder for all the assimilated clients.

The best way to create a source folder is to set up a single client machine just the way you like it and then copy that machine's entire hard disk up to the file server. While there are some caveats with this method, it is a good way to start.

When you copy up the hard disk to the file server you will notice that all of the aliases on the server still point back to the hard disk. This is obviously incorrect. You need to fix those aliases so that they point to the appropriate item inside the source folder. That way, when Assimilator copies the alias down to the destination disks, it will fix the aliases.

Assimilator User and Source Folder Permissions

When Assimilator runs on the client machines, it needs to mount the server that contains the source folder. To do this it needs an **username** (and **password**) that has sufficient privileges to see the source folder. It is strongly recommended that you give Assimilator its own dedicated username (normally "Assimilator" but you can give it any name you like), and that this user be highly restricted. Specifically the Assimilator user should not have any permissions to write to any volumes on the file server. Also the user's password should not be the same as any of your administration passwords.

Note: The Assimilator user's password is inherently insecure because it is stored inside the Assimilator on each of the lab machines. Although the password is scrambled, to prevent idle snooping, it is not effectively encrypted.

You should now set the permissions on the source folder so that the Assimilator user can see all of the enclosed files and folders. The best way to do this is to make yourself (the administrator) owner of the folder -- with See Folders, See Files and Make Changes privileges -- and make the Assimilator user the group of the folder -- with only See Folders and See Files privileges. Don't forget to make all of the enclosing folders the same as this, otherwise you'll end up with a half-assimilated machine!

Now that you've created the source folder and the Assimilator account you should fill out the items in the Source panel of the customised Assimilator's window. In preparation for doing this, mount the file server using the Assimilator user's account and password and check that the user can see all of the items within the source folder. Then click button in the source panel and use the standard directory dialog to set the source folder. Then you should fill in the Username and Password fields with the Assimilator user's name and password.

Note: If you leave the username field blank then Assimilator will attempt to log in to the file server as a guest.

Ready To Run

At this stage you can save your customised Assimilator as either an application or an extension and use it in action. In fact, it might be a good idea to try it out. Just take a copy of the Assimilator to a lab Mac and run it. It should automatically mount the file server, synchronise the Mac's hard disk with the source folder and then restart the machine. If not you should consult the check list at the end of this section.

Note: Assimilator virtually always restarts the machine at the end of an assimilation. This is because it has almost always downloaded new versions of some files and the Mac must restart in order for these to take effect. You can use the 'Don't Restart Flag' if you want to force Assimilator not to restart, but normally it will make the right decision about restarting.

WARNING: If you save this customised Assimilator as an application then running it is extremely dangerous, because it will automatically start Assimilating the local hard disk. Do not double click it to edit it, because that will run it! Instead, use Open from Assimilator Admin's File menu to open the Assimilator.

After you have tested your newly created Assimilator, you might want to customise it further. To do this, use the Open command on Assimilator Admin's File menu to open the Assimilator. There are a number of extra configuration options you might want to explore. These are discussed in the next section.

Check List

These are the steps required to create a minimal customised Assimilator. If you have troubles getting it working then make sure that you have performed each of these steps.

1. Create a source folder by dragging a copy of the lab machine's hard disk on to the file server.
2. Create an Assimilator user on your file server.
3. Set the privileges of the source folder so that the Assimilator user can see all of the folders and files within the source folder.
4. Mount the server volume containing the source folder using the Assimilator username and password.
5. Check that you (that is, the Assimilator user) can see the source folder.
6. Run Assimilator Admin and create a new untitled Assimilator.
7. Set the source folder by clicking on the button in the source panel.
8. Set the Assimilator username and password using the items in the source panel.
9. Set the Kind to Application.
10. Save the Assimilator.

Further Configuration

Once you have got a working Assimilator you can continue to configure it. There are a number of configuration options you might want to explore in order to gain better security and efficiency.

Abort Password

When Assimilator is assimilating a client machine, it displays a dialog with a Stop button. It's obvious that a student in the laboratory should not be allowed to stop the assimilation, so the operation is password protected. You can set the **abort password** in the Miscellaneous panel of the Assimilator's configuration window.

When Assimilator is running and a user clicks the Stop button, they are presented with a dialog asking for the abort password. If they enter the correct password then Assimilator stops. If they enter the incorrect password then the assimilation continues. Finally the password dialog times out after 30 seconds, so the assimilation continues if no password is entered.

Minimum Free Space and Trash Management

The Miscellaneous panel also holds an item where you can set the **minimum free space** that Assimilator should guarantee. At the end of the assimilation, Assimilator will delete items in the Trash to ensure that this amount of space is free on the destination disk. Normally you would set the minimum disk free space such that there is enough space for standard user operations, such as editing a large word processor document. Assimilator defaults to a setting of 2MB.

Leaving the minimum free space low is good because, if a user accidentally saves a file on the local hard disk of a machine, it will stay around in the Trash for a while. On the other hand, if you are a particularly fascist system administrator, you can set the minimum disk free space to a very large number, thereby ensuring that any files thrown away by Assimilator are automatically deleted.

The exact trash management scheme it uses is described in the following paragraphs.

Under normal circumstances, Assimilator puts any files it wishes to discard into the Trash. Obviously this process cannot continue indefinitely; sooner or later the disk will fill up. So Assimilator is forced to make decisions about which files to delete. It does this by deleting the oldest files first. To do this it uses a sophisticated Trash management scheme.

When it commences an assimilation, Assimilator creates a folder in the trash called "Assim ddmmyy hhmm". It then puts any file that it needs to throw away into that **Assimilator trash** folder. This means that all the files in the Trash are recorded by the time and date when they were assimilated.

When Assimilator discovers that the disk is too full it starts deleting files in the Trash. It progressively deletes the files in the oldest Assimilator trash folder in the Trash until there is enough space free. If deleting all of the Assimilator trash folders fails to yield enough disk free space then it starts deleting files at the top level of the Trash; these are normally the files that users have discarded.

Minimum Rerun Time

Just like a television station, Assimilator has a minimum time before it will run the same program again (: You can set this time **minimum rerun time** , in minutes, in the Miscellaneous panel of the Assimilator's configuration window. The default is 10 minutes.

This rerun time has a number of consequences. If you have Assimilator set to run at startup, either by putting an Assimilator application in the Startup Items folder or by putting an Assimilator extension in the System folder, then you must be careful to set the minimum rerun time to longer than the time taken for the machine to assimilator and then restart. If the time expires before Assimilator has had time to run, assimilate and restart the machine, then it will notice that the time has expire and automatically run again, looping forever, or until an exasperated system administrator fixes it.

At times it may be useful to set the minimum rerun time to 0 minutes. One common technique for setting up a lab is an Assimilator extension that runs at startup time to clean up the machine when it restarts and to have an Assimilator *application* (with an obvious name like "Clean Hard Disk") on the desktop. This way, if a user encounters a machine that is broken they can fix it by running the "Clean Hard Disk" application, somewhat more obvious than restarting. You could also imagine a situation where you don't even have a copy of Assimilator in the Extensions folder and all hard disk cleaning is done under user control.

If you hold down the control key while launching Assimilator, it will run regardless of the minimum rerun time. This is particularly useful for testing changes to your source folder.

Note: The time of the most recent assimilation is stored in the Backup date of the Assimilator application or extension. If you use a backup program that sets this date, Assimilator may run more than required. Also, if Assimilator downloads a new copy of itself (typically because you modified the version in the source folder), then the new version will not have been run and it's backup date will be unset. This means it will always assimilate, even if it's just to set its own assimilator time. Under certain circumstances it is possible for Assimilator to run *three times* before things settle down.

The Database of Deviancy

The goal of Assimilator is to make all of the Macintoshes in the lab look identical. This is a worthy goal but, as we all know, too much of the same thing can be bad. Some settings, such as the Sharing Setup Macintosh Name, must be different for each machine. This section describes the database that Assimilator uses to set machine specific parameters on each lab machine.

Database Concepts

There is database stored in each Assimilator document, application and extension. You can edit the database by opening the Assimilator with Assimilator Admin. The database comprises a number of records, with each record containing the following fields:

Identify By
This is the key field. It determines whether Assimilator should consider this record when it's assimilating a machine. It contains either an Ethernet address, a hard disk creation date or a machine ID file.

Machine Name
This field contains whether Assimilator should set the Machine Name and, if so, the value to which it should be set. The Machine Name is normally set by the Sharing Setup control panel and stored in the System file and the Users & Groups Data File.

Owner Name
This field contains whether Assimilator should set the Owner Name and, if so, the value to which it should be set. The Owner Name is normally set by the Sharing Setup control panel and stored in the System file and the Users & Groups Data File.

Volume Name
This field contains whether Assimilator should set the name of the destination volume. There are three options here: Assimilator can preserve the existing name, set it to the

name of the source folder or set it to some value held in the database.

TCP IP
This field contains whether Assimilator should set the MacTCP or OpenTransport IP number. Normally the IP number is set via the MacTCP or TCP/IP control panel and the value is stored in the appropriate preference file.

Configure Group Membership
This is covered in detail later in the Customization Toolkit: Groups section. Briefly, Group membership permits you to customise which versions of a file (or folder) a particular machine will receive on a machine by machine basis.

When Assimilator runs, it walks down the records in the database looking at the Identify By key field. If this field matches the machine on which Assimilator is running then it uses that database record to set the various machine specific parameters. If no records match then Assimilator uses the **<Default>** entry, which is always present in the Database.

Note: Assimilator sets most of these preferences by bashing the relevant files with great violence. A number of these bashes have certain compatibility drawbacks. Firstly, setting the Machine Name requires Assimilator to hack up the Users & Groups Data File, because a copy of the machine name is held in that file and there in no sanctioned way to modify it. Secondly, setting the IP number requires directly bashing the TCP preference files because there is no sanctioned way to do it otherwise. Assimilator will be revised to fix any compatibility problems that arise from future system software. Welcome to the world of MacLab hackery!

The Key to Identify By

There are three ways to identify a machine uniquely:

Ethernet Address

Every Ethernet card (or Macintosh with built-in Ethernet) must contain a 48 bit number that identifies the card *uniquely* . If your Macs have Ethernet then you definitely should use the Ethernet address to identify them.

Hard Disk Date
The creation date on the destination hard disk is a common way of identifying hard disks uniquely. It works because the creation date is stores in seconds and it's difficult to get two disks with the same creation date if you initialise them by hand.

Unfortunately if you get a big batch of machines -- such as a new Mac laboratory for example -- you will often find that all the hard disks have the same creation date. If you reinitialise each of the hard disks manually then the creation dates will be unique and you will be able to use them as the key.

Machine ID File
The final, and least preferred, mechanism to identify a machine is by a machine ID file. This is a file in the Preferences folder whose name uniquely identifies the machine. Assimilator will create this file in the System Folder of the Lab Machine when you chose this option while doing a Quick Add (see below 'Adding Entries Quickly'). If you wish to generate these files yourself, they must be of type

```
'Asim'
```

and creator

```
'Asim'
```

, with zero length data and resource forks. Assimilator ignores such files when assimilating the hard disk. You must be extra careful to make sure that your machine ID files meet these requirements, otherwise they'll be assimilated.

Database Operations

Assimilator Admin contains all of the basic tools required to maintain the database. The Assimilator's configuration window contains an Edit Database button, which brings up a dialog showing each of the database records. You can click the Add button to add a new database record, the Delete button to delete the selected records, the Change button to edit a single record, and the Done button to dismiss the dialog. You cannot remove the **<Default>** entry.

When you add or change a database record you encounter the **Add/Edit Entry** dialog, which allows you to set the fields of the record. The meaning of each of the fields is described in a previous section.

Adding Entries Quickly

The most common problem you face as a lab administrator is having a whole bunch of machines whose Identify By keys (typically their Ethernet addresses) are unknown.

Fortunately, Assimilator Admin has a shortcut for adding lots of individual entries easily.

The first step is to put a copy of Assimilator Admin on a floppy disk. Now put an Assimilator document, application or extension in the same folder as the Assimilator Admin program. Now go to a lab machine and run Assimilator Admin *while holding down the control key* . Assimilator Admin will bring up the **Quick Add Entry** dialog.

This dialog has the Identify By information set up in the best way possible. If the machine has Ethernet then the key is set to the Ethernet address of the Ethernet card. If the machine does not have Ethernet then the key is set to the hard disk's modification date but you can easily switch it to a machine ID file.

After you've confirmed the Identify By key field, you can simply type in the other information associated with this database record. When you click OK Assimilator Admin will automagically add this record to the database contained in the Assimilator document, application or extension in the same folder as it and then quit. You can now eject the floppy disk and take it to the next machine and repeat the operation.

Note: If you OK the dialog when the machine ID file option is set then Assimilator Admin will create the file for you.

The Import/Export Business

Because Assimilator Admin's facilities for editing the database are rather basic, it also provides a facility to export the database to a text file and then reimport a modified database from a text file. The format of this text file is rather hard to explain- if you're interested you should export a simple database and work it out from that. But briefly, in order, the Database entries are the following tab separated fields:

Version - the version of the database format, not Assimilator.
Ident By - 0, 1, or 2 meaning Ethernet Address, Harddisk Date or Machine ID File respectively.
Ethernet Address - the hardware ethernet address. May be blank.
Harddisk Date - a signed 32 bit number (it might be negative). Generate these using Assimilator, or preferably leave it as 0 and use the Ethernet Address.
Machine ID - the machine ID file name. May be blank.
Machine Action, Machine Name - 0 to preserve, 1 to set.
Owner Action, Owner Name - 0 to preserve, 1 to set.
Volume Action, Volume Name - 0 to preserve, 1 to set to the Server Folder Name and 2

to set to the specified name.
TCP action, TCP number - 0 to preserve, 1 to set.
Groups Database - a string of diamond separated groups.

Fine Tuning

Once you have got Assimilator working then you can spend considerable time fine tuning it for your particular installation. This section describes some of the useful mechanisms for doing this.

Assimilator Log

When Assimilator runs it creates a log of all its actions in a file called "Assimilator Log" in the Assimilator trash folder. You can look in this log for a description of the actions taken and the reasons for those actions.

Interpreting the log is the key to fine tuning Assimilator. Each line in the log starts with an action and ends with a reason. For example, imagine some has thrown away the ClarisWorks application on the hard disk. When Assimilator runs it sees that the ClarisWorks application is missing and downloads a new copy from the server. The log entry would look like:

Downloading "ClarisWorks" because it is missing.

The log actions are:

Startup
Assimilator records the date and time that it runs as the first item in the log.

Finish
Assimilator records the date and time that it finishes as the last item in the log. You can use this to calculate the time taken for assimilation.

Trashing
Assimilator is throwing away an item; the reason is given as part of the log entry.

Downloading

Assimilator is downloading an item; the reason is given as part of the log entry.

Blessing
Assimilator will automatically bless any System Folder that it downloads and record that action in the log. The exact criteria for a System Folder to be blessed is covered in a later section.

Deleting
Assimilator is deleting an item. The primary reason is given as part of the log entry. Items are only deleted if they cannot be held in the trash while maintaining the minimum disk free space requirement.

The log reasons are:

it is missing
The item is present in the source folder but missing on the destination hard disk.

one is a folder, one is a file
The item is a file on the source folder and a folder on the destination hard disk or vice versa.

moddate
The item on the destination volume has a different modification date from that in the source folder.

file type
The item on the destination volume has a different file type from that in the source folder.

file creator
The item on the destination volume has a different file creator from that in the source folder.

rsrc length
The item on the destination volume has a different resource fork length from that in the source folder. Basically this means that the file has changed size.

data length
The item on the destination volume has a different data fork length from that in the source folder. Basically this means that the file has changed size.

it is marked to always download
The item on the server has been marked as "Always Download" using a Finder label.

it is in the way of a download
The item on the destination volume is being discarded because a replacement item is being downloaded. This entry always has a matching entry that explains why the replacement item is being downloaded.

it should not be here
it should not be here (root)
The item on the destination volume is being discarding because it is not in the source folder. Don't ask me why there are two different messages for this!

we are zarching the entire harddisk
Assimilator is zarching the entire hard disk and thus it discarding all files.

the desktop folder should be empty
There are items on the destination volume's desktop but the desktop is supposed to be empty.

Labelling Your Source Folder

Once you have got your machines assimilating happily you should look at your log and see what actions Assimilator is taking and why. What you will find is that a number of files are being downloaded every time Assimilator runs. This is usually because the files are modified in the process of system startup. Unfortunately the most serious offender is the System file, which is also very big. So it's worth your time to analyse the Assimilator log and see how things can be improved.

The way you stop a file being downloaded every time is to label it (using the Finder) to indicate that to Assimilator that it is to take a special action. By default the labels and there actions are:

None -- Download if Modified
Essential -- Always Download
Hot -- Download if Different
In Progress -- Download if Missing
Cool -- Never Download

Personal -- Download if Modified & Make Invisible
Project 1 -- Always Download & Make Invisible
Project 2 -- Follow Alias

You can change these default actions using the instructions in the next section but normally there is no need to do so.

The default action, which corresponds to **no Finder label** , is to download a file if it has been modified. The definition of **modified** , as far as Assimilator is concerned, is that either the file has changed or the modification date has changed.

In the case of the System file, the modification date changes every time the machine restarts. You can tell Assimilator to ignore this modification date by labelling the file with the **Hot** label, which means the file is only downloaded if it is different -- where **different** means that the file size, type or creator has changed.

If, after you've labelled it Hot, the file is still being downloaded, then you can take more drastic action. If you label the file as **In Progress** then it will only be downloaded if it's **missing** , that is the file is present on the source but not on the destination. Although this is generally not recommended for use on files it is useful for recalcitrant files whose modification date and size change at system startup. This label acts slightly differently on folders: if a folder is marked as download if missing and the folder is present on the lab machine Assimilator does not search inside that folder, it just moves on. If the folder is missing normal Assimilation occurs and the folder, along with its contents, are downloaded according to the labelling. Very useful for creating User folders.

The other labels are useful too. The **Essential** label tells Assimilator to **always** download the file every time. This is useful for files like the MacTCP DNR file, which regularly get corrupted. Because the file is small, the load from downloading it every time is minor, certainly compared to the hassle of having to deal with a corrupted one.

The **Cool** label corresponds to the **never** download action. Assimilator will never download an item that is labelled Cool but it will also not throw it away. One possible use for this is to create a "Users" folder, where users can store their own personal files. If you label it Cool then Assimilator will not attempt to download the folder from the source (very sensible because it would be empty on the server) and it will not discard the folder on the destination.

The final action is accessed through the **Project 2** label. It is the Follow Alias label. This allows you to put an alias to a file on the server in your Source Folder, and Assimilator will pretend that the alias is part of the Source Folder. Assimilator resolves the alias effectively substituting the document or folder in place of the alias (and if necessary, following into the folder and Assimilating its contents). This is an easy way to create alternate configurations, by making different Assimilators which point to different Source Folders, which are mostly made of aliases labelled with the Follow Alias colour.

The Finder labels **Personal** and **Project 1** are analogues of some of the other labels except that they also tell Assimilator to render the item invisible after it has downloaded it. This is useful in scenarios where you want to project the System Folder on the destination disk from casual tinkering. While you can't make the entire System Folder invisible (the Finder gets upset) you can make the Extensions folder, System and Finder invisible. This ensures that the casual tinkerer can't prevent a clean system restart, which in turns means that Assimilator will run and clean up the machine at that time.

Warning : Mac OS 8.x will not run Startup Items or Extensions if the System Folder or Startup Items/Extensions folder is invisible.

Note: Assimilator will download invisible files in the source folder and render them invisible on the destination disk. The invisible labels are useful when you want the files to be visible on the server (so that you operate on them, for example installing new Fonts in the System file) but invisible on the lab machines.

One helpful hint is to change the name of the Finder labels (in the Labels control panel) to correspond with their Assimilator actions.

Action Reference

Assimilator supports the following actions:

Download Always
The item (and its contents if it's a folder) is always downloaded. Any existing item on the destination is always trashed.

Download if Modified
If applied to a file, the file is downloaded if the one on the destination is different from the one on the server or the file's modification date have changed.
If applied to a folder this just causes Assimilator to recursively operate to items within

the folder.

Download if Different
If applied to a file, the file is downloaded if the one on the destination is different from the one on the server. The definition of different is the the file size, type or creator has changed.
If applied to a folder this just causes Assimilator to recursively operate to items within the folder.

Download if Missing
The item is downloaded if it is in the source folder but not on the destination disk. If applied to a folder Assimilator does not recursively operate on items within the folder unless the folder is missing on the destination disk. As a special case, if there is a file in the source and a folder on the destination (or vice versa) the item is considered to be missing.

Never Download
The item is never downloaded. If the item exists on the destination then it is preserved. If it doesn't exist on the destination then nothing happens. If there's a file on one and a folder on the other then still nothing happens.

Follow Alias
The alias is resolved and the file or folder the document resolves to is substituted in place of the alias.

Any of these actions can be combined with the invisible action, which forces the item to be invisible on the destination disk.

Customising Label Actions

There are 6 basic Assimilator actions:

Download if Modified
Always Download
Download if Different
Download if Missing
Never Download
Follow Alias

There is also one modifier, the invisible checkbox, which determines whether an item is to be made invisible after it has been downloaded. This yields 10 (2 times 5) combinations. Eight of these combinations are associated with the eight Finder labels. By default the associations are:

None -- Download if Modified
Essential -- Always Download
Hot -- Download if Different
In Progress -- Download if Missing
Cool -- Never Download
Personal -- Download if Modified & Make Invisible
Project 1 -- Always Download & Make Invisible
Project 2 -- Follow Alias

You can change these associations using the popup menus and checkboxes in the Label Actions panel of the Assimilator's configuration window. Normally the default associations are correct and you don't need to change anything.

Miscellanea

This section contains a whole pile of hints and tips that didn't fit anywhere else (:

Desktop Database

In order to support double clicking documents, the Macintosh maintains a list of applications are available in something called the **desktop database** . When it downloads or trashes an application (technically anything with a bundle), Assimilator attempts to update the desktop database. Normally this works fine and you do not need to worry about it but in certain circumstances it may be necessary to rebuild the desktop database completely. You can do this by holding down the command and option keys while the Finder starts up (usually at the end of the startup time).

There are extensions and utilities which can be used to force the Desktop Database to be rebuilt. AutoBuild lets you specify how often you would like the Desktop Database rebuilt.

Reputedly you can also create a filed called "Desktop" in the root level of your hard disk and the next time the machine restarts the Desktop Database will get rebuilt. This is an easy way to force a periodic rebuild of the desktop database.

The Desktop Folder

Under System 7, items that are "on the desktop" are actually held in a special, invisible folder called "Desktop Folder" that lurks at the root of each hard disk. You would think that they way to achieve this is to create a "Desktop Folder" inside the source folder and put the items you want on the lab machine's desktops in that folder. Unfortunately it's not that easy!

Firstly, the Finder won't let you create a folder called "Desktop Folder" because the name is "reserved by the system software". This is fairly easy to workaround. If you create a folder called "•Desktop Folder•" (immediately inside the source folder) then Assimilator will treat it as if it's the desktop folder.

Secondly, it's quite hard to get the icons to appear in the right position on the desktop. There are a number of problems with this, not all of which are well understood. The best solution is to follow the advice given earlier in this document, whereby you create the source folder by dragging the entire hard disk of a prototype lab machine up to the file server. This will create a "Desktop Folder", including all of the icons in the right position. You should then modify that folder carefully, making sure to always keep it in the big Icon view.

Blessed are the System Folders

The Macintosh keeps track of the currently active System Folder on each disk because it needs that information at startup time. For historical reasons this folder is known as the "blessed folder" and the act of setting this folder is known as "blessing a folder". Blessing the wrong folder will prevent the Macintosh from starting. In order to guarantee that the correct folder is blessed, Assimilator will bless any folder that contains a System file. It recognises the System file by its file type

```
'zsys'
```

and creator

```
'MACS'
```

. You should make sure that you only have one System file in your source folder and that it is the correct System Folder.

Assimilator adds a log entry whenever it blesses a folder. If you have blessing problems you should consult the log to find out what went wrong.

Special Files and Folders

Because Assimilator is fully System 7 aware it knows about, and deals correctly with, a number of special files that are maintained by System 7. These are documented here for completeness:

AppleShare PDS
Move&Rename
Network Trash Folder
These file and folders are found in the root directory of the hard disk and used by file sharing.

Desktop DB
Desktop DF
Desktop
These files are kept in the root directory of the hard disk and make up the desktop database.

Trash
Wastebasket
This folder holds the items that are in the trash. (Wastebasket is the British localization of Trash.)

Shutdown Check
This file is created by the System 7.5 General Controls control panel shut down warning check.

VM Storage
This file is used by the System 7 VM system.

Aliases Explained

Assimilator attempts to correct any aliases that it downloads from the source folder. This process is a little tricky and it deserves an explanation. When it downloads an alias from the source folder, Assimilator resolves the alias. If the alias points to some item within the source folder then Assimilator corrects the alias so that it now points to the corresponding item on the destination hard disk. This is usually the Right Thing .

For example, lets say your source folder is called "Source Folder" and is held on a volume called "Source Volume" and you have an alias "Source Volume:Source Folder:•Desktop Folder•:SimpleText alias" that points to the application inside source folder, that is "Source Volume:Source Folder:Applications:SimpleText". Assimilator will download the alias to "Dest Disk:Desktop Folder:SimpleText alias" and the application to "Dest Disk:Applications:SimpleText". If Assimilator took no other action than the alias on the desktop folder of the destination disk would still point to the source folder and not to the copy of SimpleText on the hard disk. This is obviously the Wrong Thing . Instead Assimilator corrects the alias so that it points to "Dest Disk:Applications:SimpleText".

Whenever Assimilator fixes an alias, it adds a log entry describing its actions. In fact, it's a bit verbose in reporting it's actions, so it will list all the aliases it looks at and will show "errors" for the ones it decides not to fix (for example, aliases pointing to somewhere outside the source folder). The rule here is "don't be alarmed" :-)

Destination Disk Choice

Each time Assimilator runs, it's faced with the choice of which disk to assimilate. Surprisingly enough, it's actually quite hard to determine this.

If you are not using Partition labelling (explained later under Multiple Volumes), then Assimilator uses the following algorithm: it assimilates the first disk that is larger than 2MB (this eliminates floppy disks), is not hardware locked (which eliminates CD ROMs) and is not an AppleShare volume (which means it won't try to assimilate your file server). You can use this algorithm to your advantage when building an Assimilator floppy, as described in a later section.

Time is Relative

It's important to remember that Assimilator uses the modification date of files to determine whether they've changed. If the times on your server, you administration machine and your lab clients are different then Assimilator may erroneously conclude that files have been modified even whether they haven't. I recommend that you install a time synchronisation utility to ensure that the time is the same on all of your machines. Time synchronisation programs include LocalTime (freeware), Network Time (shareware), LabMaster (shareware) and KeyServer (commercial).

Backdoor Reference

Assimilator has a surprising number of backdoors, that is special key combinations that cause it to perform some special operation.

control key while launching Assimilator
If you hold down the control key while launching Assimilator, it will ignore the minimum rerun time and run immediately. This is described better in the section "Minimum Rerun Time". Assimilator will also redownload all aliases - this is useful to fix any alias resolution problems caused by problems in version 2.0.

control and shift keys while launching Assimilator
If you hold down the control key and shift keys while launching Assimilator, it will 'zarch the destination disk'. This process involves throwing away all existing files on the disk and downloading all new copies from the source folder. Zarch does not delete the desktop database files or any of the other special files. Zarch is the best way to completely rebuild a suspect Macintosh.

control key while launching Assimilator Admin
If you hold down the control key while launching Assimilator Admin, it will display the Quick Add Entry dialog. This is described better in the section "Adding Entries Quickly".

Assimilator Boot Floppy

It is possible to build a single floppy disk that will boot any currently available machine and assimilate it. This task has been greatly simplified by the System 7.5 Network Access Disk that was recently released by Apple. At the time of writing, a disk image for this is available from:

<ftp://ftp.info.apple.com//Apple.Support.Area/Apple.Software.Updates/US/Macintosh/Utilities/Network_Access_Disk_7.5.sea.bin>

There are two approaches you can take but both of them start with the Network Access Disk. The first approach merely requires you to boot the machine from the network access floppy, put it on the correct network using the Network control panel on that disk, then log into the file server using AppleShare (also on that disk) and run Assimilator from the file server.

The second approach is more automated but it's also more work to set up. First you start with a network access floppy. You then throw away the Finder on that disk, which frees up considerable disk space. Then take a copy of your customised Assimilator application and change its file type to

```
'FNDR'
```

and its creator to

```
'MACS'
```

. Rename it to "Finder" and put it in the System Folder of the network access floppy. Assimilator will now automatically run when you boot a machine from that disk. Because of the destination disk algorithm (discussed in an earlier section) it will chose to assimilate the machine's hard disk and not the floppy.

Note: If your machine is on Ethernet then it's possible that the above procedure won't work because the machine has accidentally reverted to LocalTalk. You can get around this either by using the previous procedure or by including a PRAM zapping extension on the boot disk. Such a utility is available as:

<ftp://redback.cs.uwa.edu.au//ComSci/LabUtilities/SetXParam.sit>

Desktop Printers and Stubborn Applications

There are some situations where each machine must be configured slightly differently but Assimilator does not have support for these differences in the Database of Deviancy. One solution is the use of Group suffixes (explained in the Customization Toolkit: Groups section).

Another solution is to configure each machine individually, then configure your Source Folder so that Assimilator doesn't break that configuration. A good example of this is

Desktop Printers.

If you configure a Desktop Printer, copy it up to your source folder, then Assimilate it down to a lab machine, the Lab Machine probably won't be able to print.

A good general strategy when confronted by problems of this kind is to configure each Macintosh correctly, then label the pertinent files in the source folder as Download if Missing or Download if Changed. For Desktop Printers you need to label the "LaserWriter 8 Prefs" and the "LaserWriter 8" extension.

(You will also need to make sure Assimilator deletes the old Desktop Printer and empties the trash, otherwise the old Desktop Printer will still be the Default Printer... in the trash.)

This strategy is very useful for Applications which tie themselves to a machine by using the Ethernet address or equivalent when the serial number is entered. Make sure each machine is correctly configured and label the application in the source folder as Download if Missing or Download if Changed.

If this all feels too flimsy for you, you can use Leonard Rosenthol's UsePrinter OSAX to force the Default Desktop Printer, and you can use the the Customization Toolkit: Groups to download different versions of self modifying applications for each machine.

However, real world experience shows that simply configuring the machine once is surprisingly reliable. It is not obvious to a user that some portions of the machine are not being regularly updated.

Even More Miscellanea

Gee this document is getting big!

Startup Delay
Assimilator waits for a few seconds at the beginning of each assimilation to give you time to abort it without damage. This is vitally important if you accidentally try to assimilate a non-lab Macintosh.

Force Quit
Assimilate disables the Force Quit key combination (command-option-escape) while it is running.

Progress Bar
It is impossible to provide a decent progress bar for Assimilator without doubling the time it takes to assimilate. Please stop asking for one!

You Need More Power???
The next section outlines the Customization Toolkit, which provides a group of more complex features which can aid in solving tricky lab problems. However, even the Toolkit may not suffice. If you need more power for your Mac lab, try out RevRdist by Dale Talcott <aeh@cc.purdue.edu>. It's freeware (as opposed to Assimilator, which you are expected to pay for) and has tonnes of power (more power than I could ever hope to deal with). It is on the UMich archives in util/network/.

The Customization Toolkit

This section describes some of the very powerful customisation tools added in version 2.0 of Assimilator. These allow you to Assimilate multiple volumes, to Assimilate subfolders and to create custom configurations for different machines by adding Machines and files to Groups. Assimilator can also Assimilate from Local Media and Assimilate just a subfolder of the main image. Assimilator also has a 'protected execution' phase, where it prevents user intervention while it executes scripts or applications you have nominated. These functions can be used in conjunction to create truly adaptable setups.

These tools are intended to help in the following scenarios:

* Your lab machines have partitioned drives, or multiple volumes. You want to Assimilate all the volumes, not just the system volume. Assimilator supports multiple volume Assimilation.

* In a single lab you have different machines which have similar, but not quite the same, configurations. A handful need Photoshop installed, you want another group to have Mathematica, but they are otherwise identical. Group membership lets you use a single Assimilator Source Folder yet different machines download different elements from the Source Folder.

* You have a number of labs with a variety of different configurations. Some use custom System Folders, some require selected software, a couple require multiple partitions. You

also have finite space on your server. Using the Follow Alias colour and Groups you can create a variety of different Source Folders without duplicating a single file.

* You use Assimilator to configure new machines, which may not yet be on the network. Assimilator now lets you plug in a hard disk and Assimilate from the (local) hard disk: you don't need to go across a network.

* You don't want to use Assimilator to maintain the entire machine, you just want the Applications folder to be cleaned out and updated every now and then. Assimilator can be used to Assimilate only a subfolder of the source folder.

* You have a simple AppleScript which customises the machines in a way that Assimilator can not. Assimilator's protected execution phase lets you execute the script without worrying that users will interrupt its execution. You can also use Assimilator to start up Login software.

Groups

The single most powerful new tool in Assimilator is Group membership. Machines can be assigned to multiple different groups, and individual files tagged with multiple different group names, Assimilator will then selectively download these files according to their group membership.

If the file is tagged with a group which the machine also belongs to, then the file is downloaded. If the file is not in any of the groups that the machine is a member of, then the file will not be downloaded.

A group is simply a case insensitive text name. A machine's Group membership can be edited where you edit the machine's entry in the Database of Deviancy. In the bottom left of the machine's Entry window is the Configure Group Membership button.

A file can be added to a group by editing the file's name in the Finder and appending a list of groups separated by a '◊' character (a hollow diamond, option-shift-v). For example:

```
Eudora Pro◊Net◊Admin
```

...marks the Eudora Pro file as belonging to the 'Net' and 'Admin' groups.

Note, that since the group names are appended to the name of the file, it is a good idea to keep group names fairly short. Folders and Volume Source Folders (see below) can also be tagged to belong to different groups.

When the file is Assimilated down, the Group tags are stripped from the name, so:

```
Eudora Pro◊Net◊Admin
```

becomes:

```
Eudora Pro
```

Note, that since the group tags are stripped from the name you can have multiple different versions of the 'same' file in the Source Folder. For instance if each machine belongs to its own group called Mnnn, where nnn is a number, then you could put the Files:

```
TCP/IP Preferences◊M001
TCP/IP Preferences◊M002
.
.
TCP/IP Preferences◊M999
```

..into a folder, and each of the 999 different machines would end up with a different version of the TCP/IP Preferences file.

More Complex Group Memberships
Group membership in Assimilator is actually even more powerful than that, but from here on in it starts getting more complex and confusing. You probably won't need to use most of the additional functionality described here, so you may not wish to read further.

Machines and Files can also be excluded from groups by inserting a tilde ('~') between the diamond and the group name. So the file:

```
Myth◊~Principal◊Student◊Admin
```

...belongs to the 'Student' and 'Admin' group, but is specifically precluded from being downloaded to any machine in the 'Principal' group.

You can also leave the final group name blank (or only use a tilde) and this is equivalent to saying 'any group' (or 'not any group'). So, for instance:

```
Myth◊Admin◊~
```

...indicates only machines in group Admin should download the file, all others are specifically excluded. This is, in fact, equivalent to:

```
Myth◊Admin
```

As soon as there is a diamond in the name a machine must explicitly belong to a group before the file will be downloaded (remembering that a final '◊' is a wild card group).

The order of the group names in the file is important. The first group name on the file is checked against all of the machine's group memberships before moving on to the next group name on the file if no match is found. So if a machine belongs to groups:

```
A B C
```

And the file is:

```
MyFile◊~C◊
```

..then the file will not be downloaded because the '~C' will be matched first and the file excluded. (Remember, the final ◊ indicates 'match all machines'.) A machine which belonged to the A group, but not the C group **would** download the file.

Multiple Volumes

Many computers have multiple partitions, volumes or external drives. Assimilator can now fully Assimilate these machines.

Configuring the Source Folder for machines with multiple volumes is almost the same as configuring Assimilator normally. Create a Source Folder for each volume on the server (called a Volume Source Folder). Then drag each these Volume Source Folders into one Big Source Folder. Select the Big Source Folder in Assimilator Admin when you specify the Source Folder (that is, when you click on the Source Folder button in Assimilator Admin, select the Big Source Folder, not the Volume Source Folder).

You now need to label each of the Volume Source Folders. Since the Volume's name can be changed by a malicious student, Assimilator selects where to Assimilate each Volume Source Folder on the basis of size and whether or not the volume is the Boot (System) volume.

Each Volume Source Folder is labelled in the form:

Nameßomn

Where:

Name is what the Volume name will be set to
ß is the Beta Symbol (option-s on the keyboard)
o is the Order in which to Assimilate each of the partitions. Order numbers do not need to be contiguous
m is a magic label (actually case insensitive), one of these:

- **B** : System Volume (the one we booted off)
- **L** : Nth Largest partition (optional parameter for N defaults to Largest).
- **S** : Nth Smallest partition (optional parameter for N defaults to Smallest)

n is an optional extra, a number (for the Nth largest or Nth smallest volume)

The order of the Volume Source Folders is significant. The first Volume Source folder (the one with the lowest Order number) will get the chance to apply its rule first, followed by the next lowest Order number and so on. Note, that after the first volume has been Assimilated, that volume is removed from the list of possible destinations.

Only partitions which are Read/Write, larger than a floppy disk and not a network volume are are included in the list of possible destinations.

So for example three folders with names:

```
Systemß1B
Applicationsß2L2
Scratchß6L
```

The following will happen:
* First, the System Boot volume will be called System and have the Systemß1S Source Folder Assimilated onto it.
* Then the second largest volume of the remaining volumes (L2) will be called Applications and have the Applicationsß2L2 Source Folder Assimilated onto it.
* Finally the largest of the remaining volumes (if it exists) will be Assimilated according to the Scratchß6L Source Folder. Note that the Ordering Number (6) is simply the next smallest entry- it doesn't have to be called Scratchß3L

If only two partitions exist, only System and Applications will be Assimilated to the machine. If there are six volumes on the machine, only three of them will be Assimilated.

The System volume will be labelled according to the name of the Volume Source Folder, unless you specify otherwise in Assimilator Admin in the Machine Database settings window under 'System Volume Name'. All other volume names will be set to the server volume name (ie the Volume Source Folder name).

Remember, when you configure Assimilator in Assimilator Admin you must point Assimilator at the Big Source Folder, not the Volume Source Folders.

Group labelling also works on Volume Source Folders, so you could have:

```
Systemß1B
Applicationsß2L2
Entertainmentß4s◊slack
Artsß5s◊art
Programmingß6L◊prog
```

...which will give all machines common System and Application volumes, but the other volumes will be Assimilated according to Group membership. So a machine in the 'prog' group will get the Programming volume, while a machine in the 'slack' and 'art' groups will get the Entertainment and Arts volumes (assuming the machine has at least four volumes).

Protected Execution Phase

Assimilator carefully prevents users from disrupting its work. It can provide these benefits to other Scripts and Applications by allowing you to execute them while Assimilator protects them.

You can create three folders in the "Assimilator Preferences" in the Preferences folder of the System Volume:

- "Prescripts" contains applets or applications that are executed before Assimilators begins operation. Assimilator will execute these scripts while it blocks users access and wait for them to complete and then begin operation (the scripts will not be executed unless Assimilator intends to run). This happens before Assimilator mounts the source volume, so they can be used to mount the remote server or otherwise prepare for Assimilator's operation.
- "Scripts" contains applets or applications that are executed after Assimilator completes its operation. Assimilator will execute these scripts and wait for them to complete and then unmount the source volume. This can be used to clean up after Assimilator or make apply any additional configuration you require such as mounting public volumes (although you might have to manually unmount the volume to do this).
- "Startup Items" contains applets or applications that are executed before Assimilator terminates. If Assimilator is not going to shutdown or restart the Mac, it will relaunch the Finder and then launch each of these applications without waiting for them to complete. This can be used to launch login systems or commonly used applications or perform any additional actions.

The "Prescripts" and "Scripts" applications are only run if an assimilation occurs. The "Startup Items" only run if the system will not shutdown or restart.

Warning : The applications and scripts in "Prescripts" and "Scripts" **must** quit - Assimilator will not continue until they do.

Warning : The applications and scripts in "Prescripts" are executed before Assimilation, so this may allow users to tamper with your setup if this folder is not controlled.

Note: You can use the AppleScript command "setstate" from a Prescript or Script to change the displayed state message in the Assimilator's window.

Subfolders

You can now have custom Assimilators which only Assimilate a subfolder of the Source Folder. In Assimilator Admin first specify the Source Folder as normal, then click the Subfolder checkbox which will permit you to select a subfolder in the Source Folder. Assimilator Admin does check that the subfolder appears to be contained in the Source Folder, but it does not check for things like 'Follow Alias' colours, so it may warn you unnecessarily.

If the subfolder is not contained in the Source Folder, Assimilator will fail.

You need to configure a Source Folder so that Assimilator knows where to put the Subfolder. Assimilator will create whatever folders it needs to put the subfolder exactly where it is specified by the Source Folder on the server. This may be unfortunate if, for instance, a student renames one of the folders between the root of the volume and the subfolder.

You may need to use the Volume Source Folder labelling scheme (see above) to specify which volume you want the subfolder to go onto. By default, Assimilator will try and put the subfolder on the System volume.

Assimilating subfolders, in combination with the 'Don't Restart' checkbox may be a handy way of tidying up Application directories.

Assimilate From Local Media

Finally, Assimilator knows how to Assimilate from local media. This may be useful for

Software Installations.

When you specify a Source Folder on a local volume, Assimilator will check whether you want to Assimilate from local media. When you run an Assimilator which is configured to run from a local volume it will look for local volumes of the appropriate name, but will not try and mount any volumes across the network.

All the above functions (Subfolders, Groups, Partitions, Protected Execution, etc...) will still work when you are Assimilating from a local volume.

Conclusion/Confusion

It is *very* easy to get confused when trying to set up Assimilator. The whole issue of maintaining Macintosh laboratories is rather complicated and Assimilator is a merely one step on the path to a solution, not the entire solution.

Assimilator is the product of years of experience running Mac labs and it works well *if you set it up correctly* . When you're confronted by problems please take the time to read this manual carefully. It contains a wide array of hints and tips for setting up your lab.

Up to date information on Assimilator is maintained on the Assimilator home page:

<http://www.assimilator.com>

This includes Frequently Asked Questions, Documentation, Registration information, download links for the latest version of Assimilator and additional notes and links relating to Lab Management. If the web site does not help you can mail the support address at:

<assimilator@assimilator.com>