



# Under the Hood of Notes Client Upgrades

# Contents

Under the hood of Notes client upgrades .....	3
1. The HCL Client Upgrade Documentation: We both agree and beg to differ .....	4
2. Let's see where companies are coming from, or: some introductions first.....	5
2.1. Scripting: 30% of your pain explained .....	6
2.2. The second 30% .....	7
2.3. The remaining 40% - failed upgrades .....	8
3. Now for the "Under the hood" stuff .....	8
3.1. Detecting the current client setup details – the biggest contributor to your success.....	9
3.2. Intelligent upgrade package distribution .....	10
3.3. Performing any upgrade from start to finish silently and with ease.....	10
3.4. Getting Elevation right .....	12
3.5. A desired change of folder locations .....	13
3.6. About Operating System File System Rights.....	14
3.7. Working "with" Virus Scanners, or: Extended Validation Certificates.....	15
4. Summing it up: Getting your upgrade right in one go. For *all* your users .....	15
4.1. Further Reading .....	15
Appendix A: How MCUpgrade.exe works in detail.....	16
Appendix B: How MCUpgradeMain.exe works in detail.....	17
Appendix C: Further details on the MCUpgrade Service .....	18
Disclaimer .....	19



## Under the hood of Notes client upgrades

This document describes the common challenges of a Notes Clients upgrade. It also describes howMarvelClient (paid or free) solves these problems to make your next upgrade as fast and easy as never before. Learn more about it here:

<https://www.panagenda.com/kbase/display/mc/Creating+your+first+Upgrade+in+20+minutes>

**MarvelClient will save you time, frustration, failures and money.**



# 1. The HCL Upgrade Documentation: We both agree and beg to differ

Notes client upgrades can be easy or very much a challenge.

They're easy when you use MarvelClient [Essentials](#) (Free or [Basic](#)) together with MarvelClient Upgrade ([Free or Full](#)). But, they are a challenge when you don't.

HCL has created good upgrade documentation on all the options for upgrading your Notes Clients: [https://help.hcltechsw.com/domino/11.0.0/admin/inst\\_installingnotesclients.c.html](https://help.hcltechsw.com/domino/11.0.0/admin/inst_installingnotesclients.c.html)

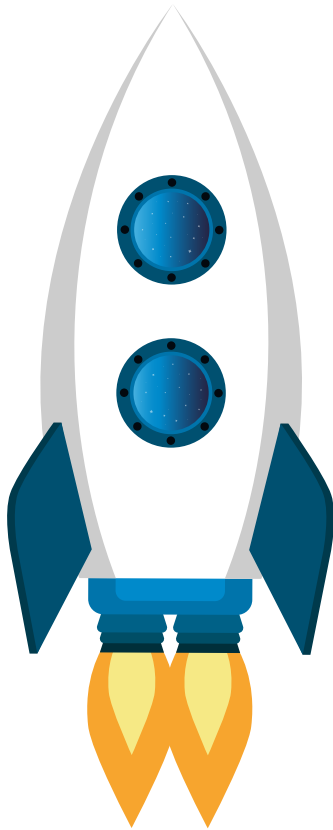
However, in our opinion, the documentation presents the following three problems:

1. The many possible different options, ranging from manual installations and upgrades, through SmartUpgrade and NotesAutoUpdate, to InstallShield Tuner and custom upgrade scripts are both overwhelming and each one introduces its very own complexities and challenges.
2. In the end, all options assume that little more than running an installer will do the job. That's far from the truth, as we will clearly show on the following pages.

3. A lot of tasks remain unmentioned, all of which are difficult or impossible to carry out without an additional software solution like MarvelClient Basic. Some of those tasks include:

- Cleaning up historically grown data directories, like templates and the workspace folder.
- Removing hard-coded server IP addresses from personal address books.
- Fixing any issues with your current deployment.
- Running maintenance on local system databases.
- Defragmenting mechanical disks on end user workstations.
- Optimizing Windows Defender configuration.
- Reviewing system requirements (we think it's much better to check whether all your clients are up to spec before even starting your upgrade.)

**Note: Together, MarvelClient Basic and Upgrade can help with *\*all\** of the above topics.**



There are also a couple of topics that we disagree on when it comes to the HCL's documentation for upgrading your Notes clients:

1.) We would never recommend to "Close all open applications, including anti-virus software." before starting your upgrade. If anything, we recommend to fine-tune your anti-virus software to exclude, for example, .nsf files and more. Additionally, you can leave almost all applications running, except for applications that use Notes, like telephony solutions and others.

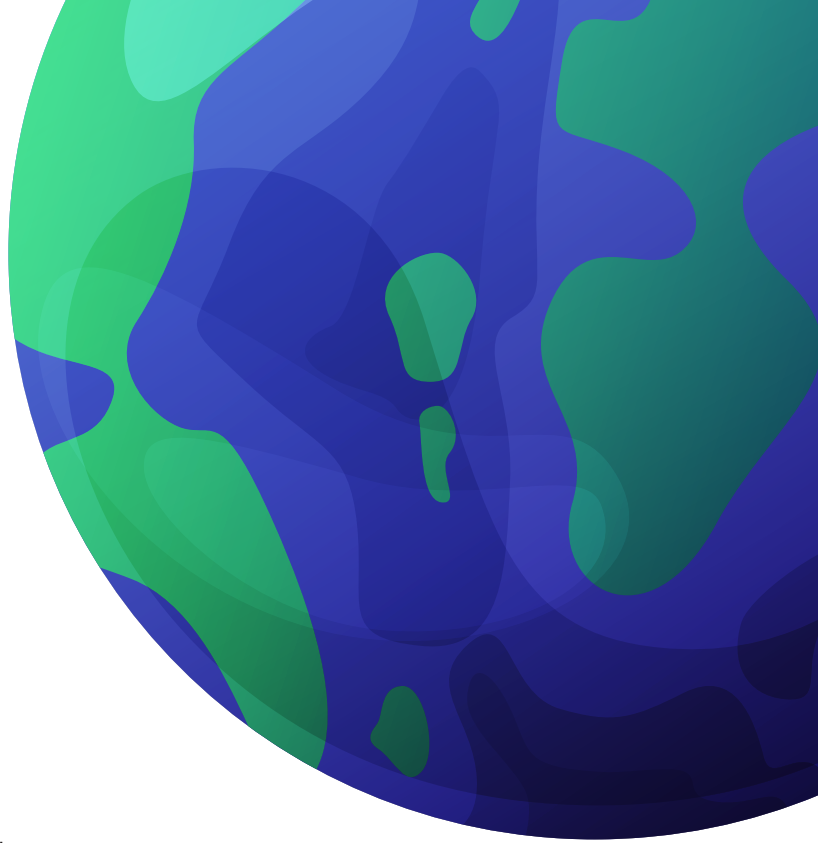
2.) Both for backup and roaming purposes, we strongly recommend MarvelClient Roaming over Notes Roaming. MarvelClient requires between 100 KB and 2 MB to roam an entire client configuration; Notes Roaming an average of 40 MB per user.

**MarvelClient Upgrade Free and Full replace the need for writing scripts, customizing installers, and trying your luck with SmartUpgrade and/or Notes Auto Update. The built-in wizard condenses our decades of experience upgrading Notes into a few clicks for you.**

## **2. Let's see where companies are coming from, or: some introductions first**

Even after excluding Administrators and Developers, most companies do not have the same notes client setup for all their end users. They often don't even know what setups are being used. From experience with our customer base (12 million licenses in over 70 countries) we know that only a handful start from a position where every end user has the exact same Notes release and folder locations.

Here are common variables that can change from setup to setup:



- Different major and minor releases
- Different fixpacks
- Different folder locations, ranging from network vs. local, through C:\Lotus\\* and C:\IBM\\* folders, to %localappdata%\Lotus or IBM\Notes\\*
- Different operating systems like Windows XP, 7, 8 and 10

Add Citrix, Windows Terminal Server, and Virtual Desktop Infrastructure (VDI) to the mix and you quickly understand how complex Notes client deployments can become.

What's more, VIPs in the enterprise often have a non-standard setup (think board or management). They are most likely to suffer when the next upgrade works for everybody but them.

---

## 2.1. Scripting: 30% of your pain explained

To get a Notes client upgrade right, most companies invest a significant amount of time in scripting around their many different client setups:

```
If data_directory Like *\Lotus\Notes\* Then
    ;this
Else
    ;that
End If
```

The next time, a new setup is discovered, the script is "enhanced":

```
If data_directory Like *\Lotus\Notes\* Then
    ;this
Elseif data_directory Like *\IBM\Notes\* Then
    ;other
Else
    ;that
End If
```

The above examples only illustrate working with different folder locations. They do not even begin to discuss detection of existing client releases, location of notes.ini files (what if there is more than one?), basic vs. standard clients, fixpack levels, languages installed, etc.

The following table summarizes the minimum number of variables defining your Notes client setups:

Notes Client Install Options	Old vs. New Install Type	Operating Systems	Notes Client Releases	Notes 10.0.1 Installers	Addons
5	4*	4	8	34	6
Single, Multi, Basic, Standard, Admin&Designer	Multi > Multi, Single > Multi, Single > Single, Admin&Designer	Windows 7, Windows 10, both32 & 64 bit	< 9 9.0.1++ 10.x 11.x, 12	One for each language, Language (MUI) Packs, Fix Packs	Connections Sametime SwiftFile JVM Patch MUI Packs

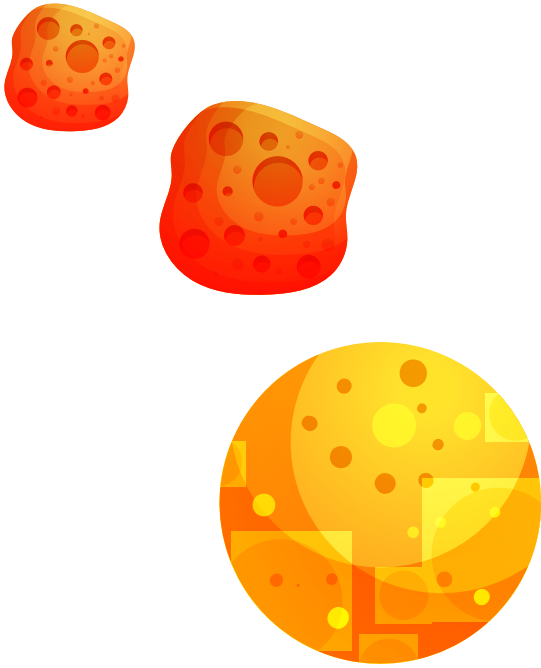
\* Don't even think about Multi > Single. Where should all the previous data directories go?

The above table represents thousands of possible combinations you could encounter. In practice, most companies face a few dozen of those; and they change with and after every upgrade.

Coding to keep up with a changing client landscape with **IF ... THEN ... ELSE ... is what leads to 30% of all frustration, effort and money spent.**

## 2.2. The second 30%

The other 30% of time and money spent on Notes client upgrades is because **scripts and software deployment solutions are not Notes aware.** They don't know much – if anything – about the current client installation (e.g. version, fixpacks, notes.ini, or folder locations). They lack the ability to customize the client setup (e.g. location documents in the Notes personal address book, desktop icons, local replicas, etc.); and Domino Policies are not enough to fill these gaps.



## 2.3. The remaining 40% - failed upgrades

The remaining 40% of all upgrade time and money is spent running after the installs that did not work out. If an upgrade works for 95% of all your users, it takes almost as much time again to fix the other 5%.

Don't forget: Many VIPs will be found among those with failed upgrades! They tend to have specialized installs, which aren't covered by the IF ... THEN ... ELSE ... .

**MarvelClientUpgrade helps you save time and money by removing IF ... THEN ... ELSE ... from the equation, by being fully Notes-aware when combined with MarvelClient Essentials or Basic, and by reducing the number of failed upgrades to near zero.**

## 3. Now for the "Under the hood" stuff

The following topics are commonly the ones that most heavily impact Notes client upgrades. The solutions for dealing with them are covered in chapters 3.1. – 3.6.

- 1.) Detecting the current client setup details (release, fixpack level, notes.ini, folder locations, etc.)
- 2.) Performing a silent upgrade from start to finish without \*any\* end user interaction, including Notes specific configuration like location documents, local replicas etc.



3.) Getting elevation right. Whilst corporate software deployment solutions solve this for most companies, such solutions are not Notes aware and hence fall short when it comes to 1.) and 2.)

4.) A desired change of folder locations e.g. from \Lotus\ to \IBM\, from network drive to local, or from single (C:\Lotus, IBM or HCL) to multiuser (%localappdata%\HCL\Notes\). Although a tool does exist to migrate single to multiuser installations ([https://help.hcltechsw.com/domino/11.0.0/admin/inst\\_sumumigrationtool\\_use\\_t.html](https://help.hcltechsw.com/domino/11.0.0/admin/inst_sumumigrationtool_use_t.html)), it is not easy to use and does not cover the migration of e.g. \IBM\Notes to \HCL\Notes folders for multiuser to multiuser installations. There are additional challenges posed by operating system file and folder security, which are covered in chapter 3.6.

5.) Combining solutions to all the issues above (and more) into a single, coherent upgrade that goes through in one go, for all of your users, and on the first try.

Let's look at how MarvelClient addresses these challenges.

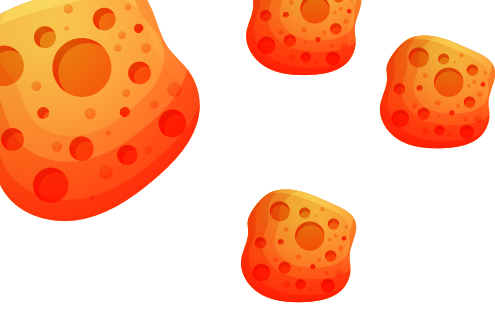
If you want to put those steps into practice, take a look at our step-by-step guide for setting up MarvelClient Upgrade [here](#).



## 3.1. Detecting the current client setup details – the biggest contributor to your success.

MarvelClient Essentials and Basic play a major role in making your next upgrade a success:

First, you can analyze your Notes client landscape to find the machines that can't be upgraded for lack of memory or disk space. It's useful to know this up front. Note that MarvelClient Upgrade automatically excludes all machines that do not meet the necessary system requirements, so you do not have to analyze anything first or at all.



Second, MarvelClient Essentials shares with MarvelClient Upgrade all it knows about your Notes clients. Release versions, fixpack level, file locations, languages, the relevant notes.ini, everything you need to know is known before you start. **This is the biggest success factor and money saver for your next upgrade: Not having to worry about all the possible setups out there, not having to use scripting to solve it manually – but simply letting MarvelClient do its magic.**

## 3.2. Intelligent upgrade package distribution

Install packages can be as large as a gigabyte or more. Many companies do not have an easy way to distribute them to everyone without bloating the network.

With MarvelClient Essentials and Upgrade, small and large files, as well as zipped files are easy to distribute. You can use network shares (UNC or mounted), web servers, or a mix of

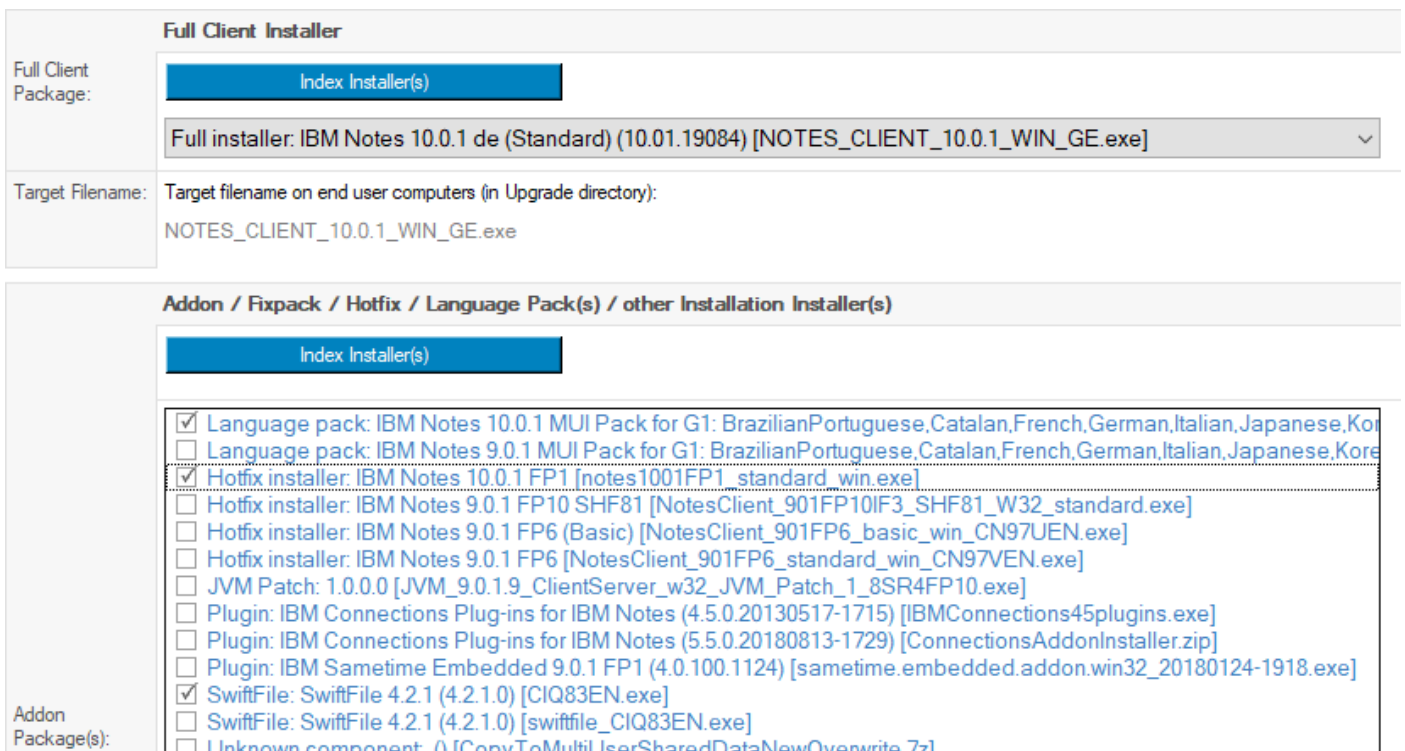
both. For example, your end users in APAC may download from share A, and users in the US download from one or multiple web servers, depending on their location.

MarvelClient Upgrade employs many methods to reduce network overload: bandwidth throttling, distribution over time, randomization of when clients download (to, for example, prevent “morning peaks”), and client attribute-based download from the nearest possible source (e.g., based on group membership, IP segment, OS settings, and more smart options).

## 3.3. Performing any upgrade from start to finish silently and with ease

There are three main issues that, combined, make upgrades tedious and error-prone:

- 1.) Using InstallShield Tuner to bake your own upgrade package(s) with multiple components e.g., when you want to upgrade to a newer release including one or multiple of the following: a fixpack, language pack(s), or other client addon packages.
- 2.) Performing in-place upgrades when installing a newer release.



3.) Any kind of end-user involvement (other than allowing end users to choose when to best upgrade for a certain number of times and/or a certain period of time).

MarvelClient Upgrade addresses all three:

First: It uninstalls your existing client before upgrading it. Whilst you can choose not to uninstall, we highly recommend using the default setting – that way, upgrades are much more successful (roughly 5-10% fewer overall errors). Naturally, it preserves each user’s existing data and settings.

Second: Installing any number of packages –

with or without upgrading the client itself – is as easy as downloading the installers, indexing them in the MarvelClient Upgrade Configuration Wizard, and selecting them:

Note: The full client package is optional, as you may just want to install one or multiple “addons”.

Third: MarvelClient Upgrade installs all selected packages in the right order, with the right parameters, and in one go.

**That way you can upgrade from any current installation to any desired configuration in a single step.**

## 3.4. Getting Elevation right

Elevation means switching to an account with administrative rights. When not using corporate software deployment, this can be challenging, especially since the introduction of User Account Control (UAC) beginning with Windows Vista.

The local built-in Administrator account in Microsoft Windows does not come with any UAC challenges, but then ideally the same password can be used for that account on all end user computers.

When using either a local (non-built-in) or Domain Administrator account, the UAC needs to be configured to allow for silent elevation when the right username and password are provided. Otherwise, end users must confirm an elevation prompt, which is unlikely to come with a high success rate.

Another option is using a local system service, which performs administrative tasks as SYSTEM.

Last but not least, an executable that requires elevation with UAC needs a special XML-manifest.

One ugly side-effect that comes with all elevation options is that usually none of them allow access to network resources, such as a Notes data directory on network drives.

MarvelClient Upgrade addresses all above issues, and more, by:

- Supporting the use of local built-in, local and Domain Administrator accounts
- Providing its own optional panagenda Installer Service\* to elevate via System/Service
- Freely combining your existing Software Deployment Solutions with MarvelClient Upgrade, as an option
- Allowing the use of multiple user account names and passwords in case there is no "one single account and password" for all computers
- Including UAC conform manifest
- Migrating Notes data directories from network resources to local (AppData) in end user context

\* Note: The panagenda Installer Service includes a variety of security-related features: RSA public/private key signatures, so called "Whitelisting" of instructions, validation of hashes of executables, Extended Validation (EV) Certificate check and more to ensure that the service is not misused. For further details, see Appendix C.

## 3.5. A desired change of folder locations

Many companies even still have the most recent Notes client releases installed in `*\Lotus\Notes\*` folders (whether that's `C:\Lotus\Notes` for single user clients or `%LocalAppData%\Lotus\Notes`). The reason for this is the use of in-place upgrades, which, apart from being less successful compared to uninstall/install, leave the data directory in its previous location.

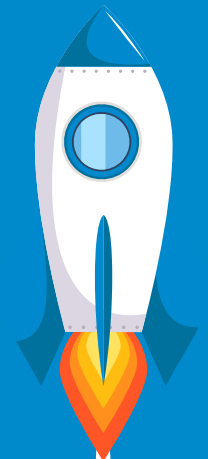
Non-standard folder locations can lead to several issues, most of which result in making it

harder for your helpdesk and IT teams to render assistance to your users.

It's about time to clean up your clients and apply best practices: Move from single-user to multi-user clients; with data directories in their proper location (`%LocalAppData%\HCL\Notes`).

Another need to change folder locations is when moving from Notes data directories on network drives to `%LocalAppData%\HCL\Notes`, which is highly recommended, even for Citrix environments / virtual desktop infrastructure (VDI).

In case you are wondering how end users can have the same data fast and lightweight beyond comparison across multiple Citrix or other VDI servers: [MarvelClient Roaming](#) to the rescue).



## 3.6. About Operating System File System Rights

Operating System file system rights are an interesting topic on their own. Things can get very interesting when, e.g., copying files from a data directory like C:\Program Files (x86)\Lotus\Notes\Data to %LocalAppData%\HCL\Notes (or even more so in the opposite direction, not because of HCL to Lotus, but LocalAppData to Program Files (x86)).

Whilst a file move takes the DACL (Discretionary Access Control List) of a file along, a file copy does not, but usually inherits the target directory DACL. If the ...Data directory does not exist yet, you can easily end up with a notes.ini file that does not give end users write access, which is critical for the Notes client to launch and function.

You might wonder “why not just move the file”? There are at least two good reasons:

1.) Most of the times you really shouldn't want to copy over the entire notes.ini. The preferred method would be to copy only what's not yet configured in the newly installed client. Plus, when changing folder locations, you must not forget to adjust notes.ini entries like Directory, SharedDataDirectory and/or NotesProgram. Speaking of which: the same applies to

template files and other historical leftovers. **A clean Notes client setup with each users' individual settings is key to a stable and performant client experience.**

2.) Sometimes the old and new data directories are on different disks, for example when moving from D:\ to C:\ or N:\ to C:\. In cases like this, you cannot move files but at best copy them over and optionally delete the source files and folders afterwards.

**MarvelClient Upgrade takes care of any change of folder locations and the setting of file system rights properly for a variety of scenarios:**

- Multi-user to multi-user migration: Automatic and smart moving of \*all\* end user data directories from, e.g., ... \Lotus\Notes\Data to ... \HCL\Notes\Data
- Moving of a data directory from a network drive to local
- Moving of a data directory from one disk/ drive to another
- If so desired, also moving the shared data directory from old to new location (we highly recommend using a clean shared data directory and adding to it as needed).

**All changes include proper adjustment of notes.ini across all migrated folders.**

## 3.7. Working “with” Virus Scanners, or: Extended Validation Certificates

Virus scanners tend to simply kill processes they consider to be potentially harmful. By itself that is a good thing. When attempting to perform the work that MarvelClient Upgrade does, however, that would definitely be bad. That is why all MarvelClient Upgrade components are signed with an official Extended Validation (EV) Certificate, which both Microsoft Windows and pretty much all serious anti-virus software vendors respect accordingly.

## 4. Summing it up: Getting your upgrade right in one go. For *\*all\** your users

MarvelClient Upgrade solves all the challenges mentioned above and quite a few more, all of which are detailed out further in Appendix A.

We hope that you, too, are enjoying MarvelClient Upgrade, be it [free or full](#) (see feature comparison at the end of the linked web page).

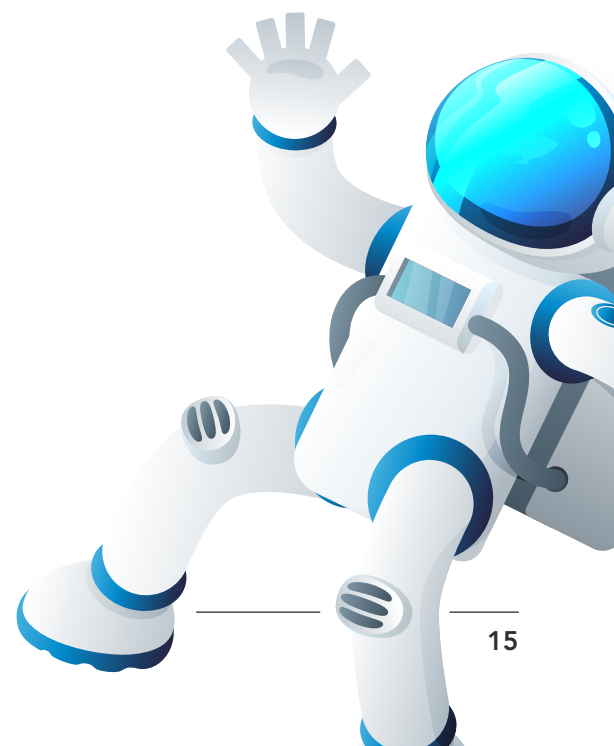
Should you have any questions or comments about this document or MarvelClient Upgrade, we are very much looking forward to hearing from you!

## 4.1. Further Reading

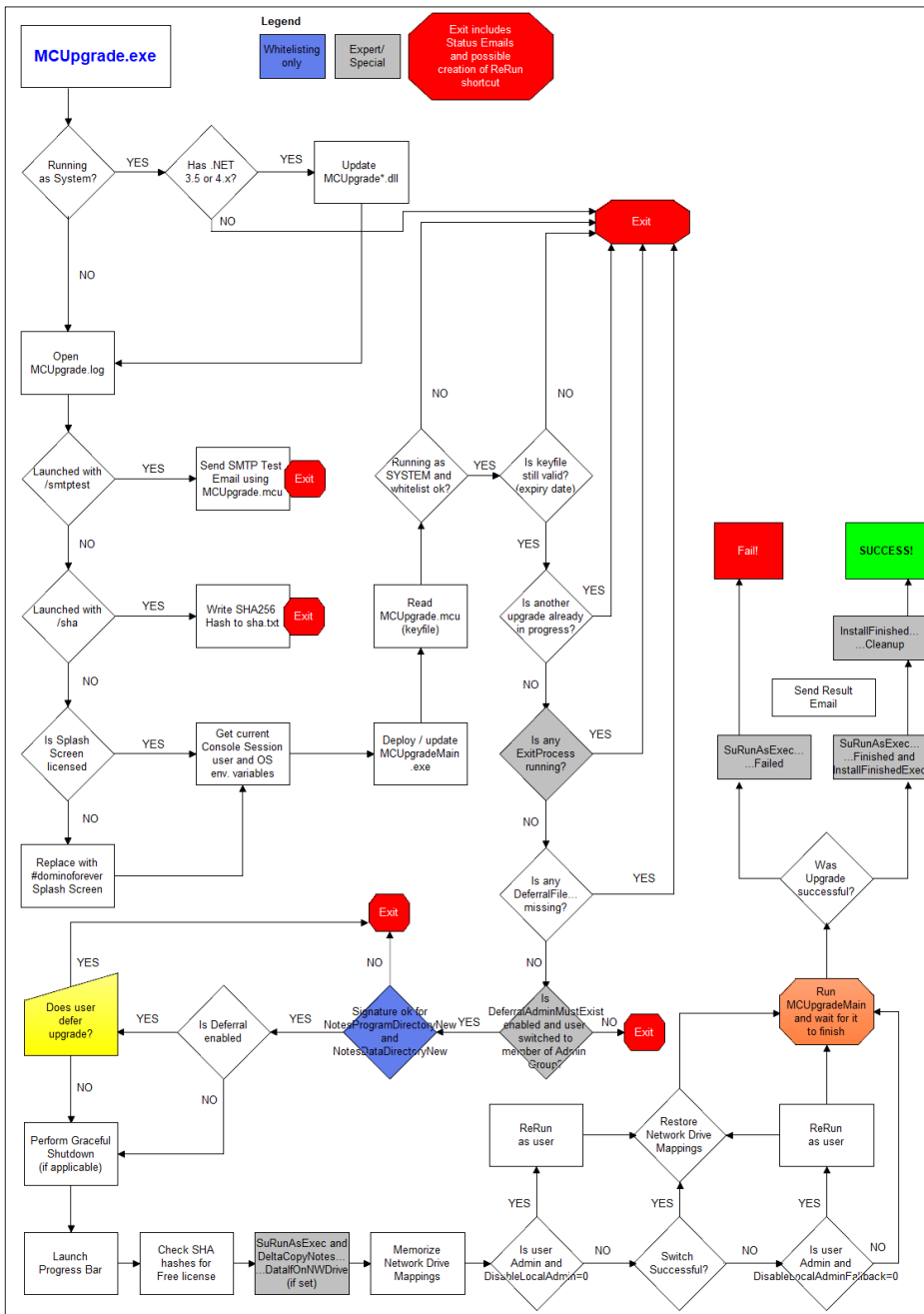
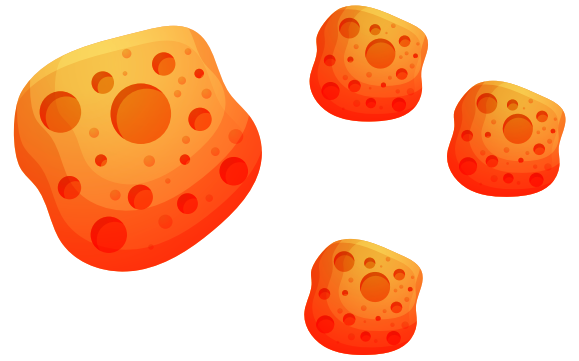
[MarvelClient Upgrade Quick Start Guide](#)  
[Taking Your First Steps with MarvelClient Upgrade](#)  
[Getting the Most From Your MarvelClient Upgrade](#)

[Get MarvelClient Upgrade Free Edition](#)

[MarvelClient Overview](#)

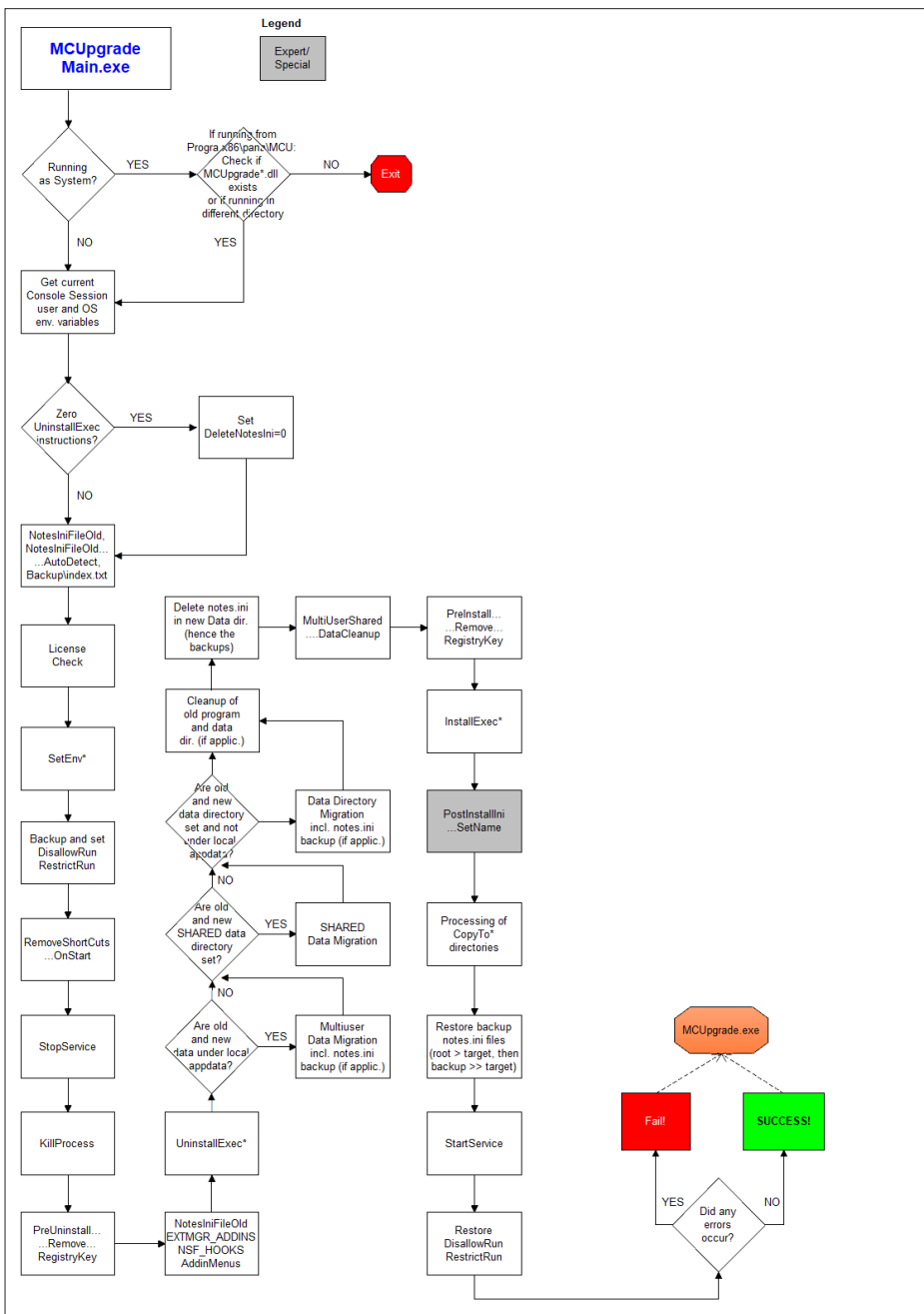
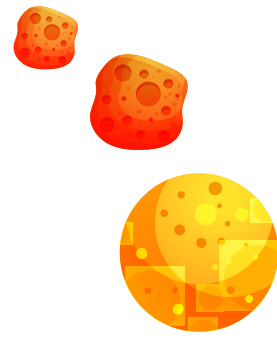


# Appendix A: How MCUUpgrade.exe works in detail





# Appendix B: How MCUpgradeMain.exe works in detail



# Appendix C: Further details on the MCUUpgrade Service

The MarvelClient Upgrade Service – also named panagenda Installer Service – and thereby elevated MarvelClient Upgrade executables have a few specialties to it that are worth noting:

1.) The service only accepts elevation requests from panagenda executables with an official and valid EV (extended validation) certificate.

2.) Both the service and thereby elevated executables require either a readily installed .NET 3.5 (MCUpgrade.dll) or 4.x (MCUpgrade40.dll) framework.

3.) When using the MCUUpgrade service, customers must create their own RSA 4096 bit keypair to lock down/secure all instructions in MCUUpgrade.ini. IMPORTANT: The private key must be handled with care = secured, just like a certifier id!

4.) The service and its components must be installed to %programfiles(x86)%\panagenda\MCUpgrade.

5.) When elevating a panagenda executable, the service copies it over into above directory and executes it therein.

6.) First MCUUpgrade.exe elevates itself into system context. Next, it updates the MCUUpgrade\*.dll files if necessary, and deploys and launches MCUUpgradeMain.exe (again into above directory).

7.) As of MCUUpgrade release >= 10.1.12, MCUUpgrade.exe will also update the panagenda Installer Service if necessary.

8.) MCUUpgradeMain.exe then

- checks the Whitelist signature
- backs up the DACL of all whitelisted UnzipArchive\* and \*Exec\* files in ini:UpgradeDirectory
- changes the DACL of all whitelisted UnzipArchive\* and \*Exec\* files in ini:UpgradeDirectory to only allow end users read access
- moves all files into above directory into a subdirectory ini:UpgradePackageName
- compares all hashes according to whitelisting

If any of the above fails, MCUUpgradeMain will exit and return an error to MCUUpgrade.exe.

9.) Any UnzipArchive\* and \*Exec\* operations will be carried out from above directory\ini:UpgradePackageName.

10.) Any other whitelisted instructions will only be executed if the respective instruction matches its signature.

**The above prevents misuse of the panagenda Installer Service.**



## Disclaimer

panagenda, panagenda product names and all related logos are trademarks owned by panagenda. All other names of products and enterprises in this knowledge base are the property of their respective owners.

panagenda reserves the right to update this document without being obliged to announce the changes or revisions.

Although all due care has been taken in the preparation and presentation of this guide, the corresponding software may have changed in the meantime. panagenda therefore disclaims all warranties and liability for the accurateness, completeness, and currentness of the information published, except in the case of intention or gross negligence on the part of panagenda or where liability arises due to binding legal provisions.

Copyright © 2021, panagenda  
All Rights Reserved.