



QUICK START GUIDE

MarvelClient™
Upgrade Free

Contents

Contents.....	2
1. Welcome.....	4
1.1. What Will You Find in This Guide?	4
1.2. What Is MarvelClient? Essentials? Upgrade? The Free Edition?	4
1.3. How do I get MarvelClient Upgrade Free Edition?.....	5
1.4. How do I get MarvelClient?.....	5
1.5. Where Can I Get Help?	5
2. MarvelClient Upgrade Wizard – Fast Track/Super Easy	6
2.1. Introduction.....	6
2.2. Download MarvelClient Upgrade Free Edition.....	8
2.3. Create an “Upgrade Configuration” action.....	9
2.4. Distribution tab	11
2.5. Conditions tab	15
2.6. Elevation tab.....	16
2.7. User Experience tab	17
2.8. (Un)Install tab	18
2.9. Tracking tab	20
2.10. Prepare or Publish Upgrade	21
3. MarvelClient Upgrade - Advanced	24
3.1. Components	25
i. MCUUpgrade.mcu	25
ii. MCKeyfileEditor.exe.....	25
iii. Whitelisting Editor.....	26
iv. MCUUpgrade.ini.....	26
v. MarvelClient Upgrade Build Directory	26
3.2. Authentication method: Local Built-In Admin.....	27
3.3. Authentication method: Local or domain users.....	30
i. Use of multiple local users	31
ii. Configuration of Windows User Account Control (UAC)	33
3.4. Authentication method: Windows Service.....	34
i. Whitelisting explained	35
ii. Preparatory activities.....	35

iii. Performing Whitelisting.....	39
iv. Whitelisting Verification (optional)	41
3.5. SMTP Server Authentication.....	42
4. How to setup the panagenda install service.....	43
4.1. Automatic Installation.....	45
4.2. Manual Installation	46
5. Security Considerations.....	47
5.1. Local built-in Administrator, local or Domain Admin	47
5.2. Using the MarvelClient Upgrade Service.....	47
5.3. Using your own corporate software deployment with MCUgrade.....	47
6. Table of Figures	49
7. Disclaimer	51

1. Welcome

Thank you for reading this Quick Start guide for MarvelClient (MC) Upgrade Free Edition!

1.1. What Will You Find in This Guide?

Section 1 is what you are reading now. It gives you a short overview of what MarvelClient Upgrade is and answers a few common questions.

Section 2 will show you how to configure an upgrade package. It will take you through all the steps needed: From downloading to configuring and preparing/publishing an upgrade package.

Sections 3 and 4 will deal with details on elevation/privileges/authentication. We provide several different methods on how to grant the needed rights to MC Upgrade.

1.2. What Is MarvelClient? Essentials? Upgrade? The Free Edition?

MarvelClient is an IBM Notes client management solution by panagenda. With it, you can centrally analyze and manage your Notes clients in various ways. Find out more about it [here](#).

Together with IBM and HCL we have recently made a subset of its functionality available to all customers using IBM Notes 9.x and newer, free of charge: [MarvelClient Essentials](#).

MarvelClient Upgrade is one of several additional modules available for MarvelClient. It allows you to perform Notes client upgrades fast, easy, and from within your Notes client infrastructure.

In order to help the Notes-world get to Notes 10.0.1, we are also making something else available for free: [MarvelClient Upgrade Free Edition](#).

It provides almost the full feature set of MarvelClient Upgrade, and is built to work with both MarvelClient and MarvelClient Essentials. 1) The only limitations are: you cannot customize the progress bar that is displayed to end-users during the upgrade and 2) There is a pre-defined list of installers you can deploy with it (Mostly Notes 10.0.1 with a few things associated with it). More details on these limits in section 2.1.

1.3. How do I get MarvelClient Upgrade Free Edition?

- If you are a MarvelClient customer, you should already have access to Upgrade Free Edition: Just perform an OnlineUpdate. Contact support if you have any issues.
- If you only have MarvelClient Essentials, [go here to request your free license](#).
- If you don't have MarvelClient yet, read the following section.

1.4. How do I get MarvelClient?

If you don't yet have MarvelClient (either Essentials or a licensed variant), don't worry. MarvelClient Essentials is freely available to current IBM Notes customers and is already included with Domino 10.

- If you have upgraded your servers to Domino 10 already, you can enable Essentials by following [this guide](#).
Notes/Domino 9 customers can download and install [Essentials from panagenda](#).
- If you are on older versions of Notes, MarvelClient Essentials is not officially supported. It can be used and should work, but without any guarantees or support.

Please make sure to create the MarvelClient databases and enable the client component as described before you proceed.

1.5. Where Can I Get Help?

Since MarvelClient Upgrade Free Edition is provided free of charge, we cannot provide any support or assistance on its use beyond this documentation. We will post several guides and articles to show how to use it on [our blog](#) over the weeks following the release.

You can, however, contact support@panagenda.com if you find any defects.

2. MarvelClient Upgrade Wizard – Fast Track/Super Easy

2.1. Introduction

The following chapter gives an overview on how to create an IBM Notes upgrade package using the brand new panagenda MarvelClient Upgrade Wizard.

Prerequisites:

- You are using either the free MarvelClient Essentials or a licensed variant of MarvelClient.
- The MarvelClient Configuration and Analyze databases already exist on at least one IBM Domino Server in your environment.
- Make sure you have the latest MarvelClient components available by running an Online Update in the MarvelClient configuration database. The local mc.dll / pmc.dll version on the user's computers **must be greater than or equal to 10.x**.
- Customers with Notes 9.0.1 FP10 IF6 must use an mc.dll / pmc.dll greater than or equal to 10.0.11 (this is because IF6 has wrong version information built-in).
- All IBM Notes clients, which should receive the upgrade package, must already be MarvelClient enabled.
- Ensure you have all needed installation packages, you plan to deploy, available on your administration computer, e. g.
 - IBM Notes 10.0.1 English (NOTES_V10.0.1_FOR_WIN_EN.exe)
 - IBM Notes 10.0.1 Multilingual User Interface – Group 1 (NOTES_CLIENT_10.0.1_ML_UI_WIN_GP_.exe)
 - IBM Swift File 4.2.1 (CIQ83EN.exe)

MarvelClient Upgrade Free will only support the following installation packages:

- IBM Notes 10.0.1 English or any other available language
- IBM Notes 10.0.1 Multilingual User Interface – Group 1, 2a, 2b and 3
- IBM Notes 10.0.1 Fix Pack 1
- IBM Swift File 4.2.1

If you plan to deploy one of the following components, a regular MarvelClient Upgrade license is required:

- Notes greater than 10.0.1 Fix Pack 1
- Notes 9.0.1 and/or MUI-Pack and/or Fix/Feature Pack and/or JVM Patch and/or Sametime embedded fix
- IBM Client Application Access
- IBM Connections Plugins for Notes
- Additional software like CTI, plugins, etc.

On the next pages, we will cover the following scenario in detail:

- Upgrade to IBM Notes 10.0.1 from any client version smaller than 10.0.1 (Notes 6.x – Notes 10.0.0)
- Adding the French, German and Japanese languages
- Adding IBM Swift File 4.2.1

Note:

Whilst MarvelClient works on Windows and Mac, MarvelClient Upgrade runs on Microsoft Windows exclusively.

2.2. Download MarvelClient Upgrade Free Edition

For MarvelClient Essentials customers:

- Get your MarvelClient Upgrade Free license key here:
www.panagenda.com/products/marvelclient/free-upgrade
- The license key must be used from within the MarvelClient Configuration database, Administration\Online Update:

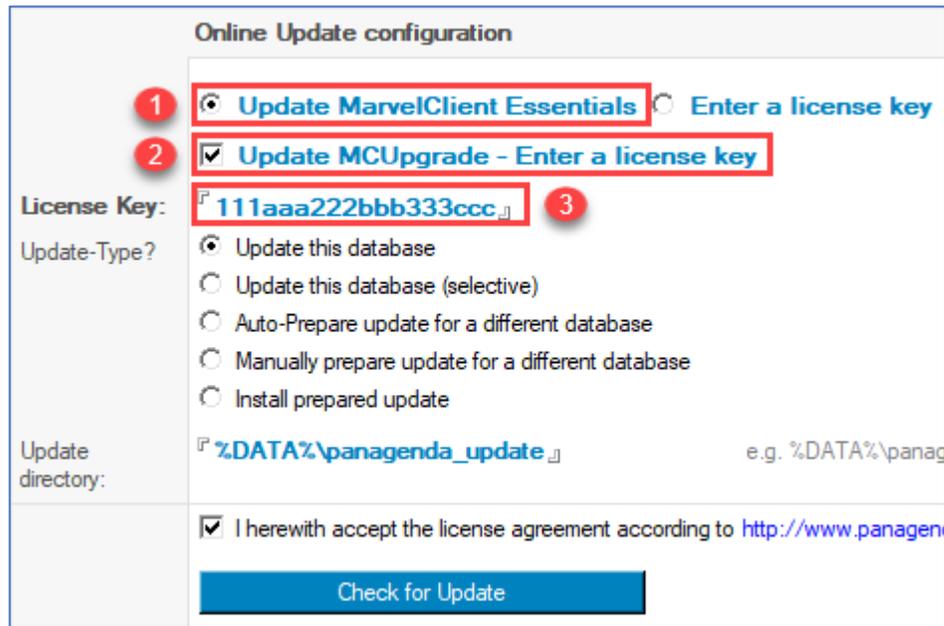


Figure 1: Online Update settings for MarvelClient Essentials customers

- If you do not see an option to enter the MCUUpgrade license key, you need to run an Online Update without the key first to update the design of the Configuration database. Thereafter, close and re-open the Configuration database and run Online Update again.
- After entering the license key, run Online Update to download the Upgrade components.

For regular MarvelClient customers:

- MarvelClient Upgrade Free has already been added to your license. Please run an Online Update to get it. You might need to run it twice; once to get the latest templates, and a second time to download the MCUUpgrade components. In case you run into an issue, please contact support.

2.3. Create an “Upgrade Configuration” action

Open the MarvelClient Configuration database, expand the “MarvelClient Upgrade” section in the navigation, open the “Configuration(s)” view and create a new “Upgrade Configuration” action.

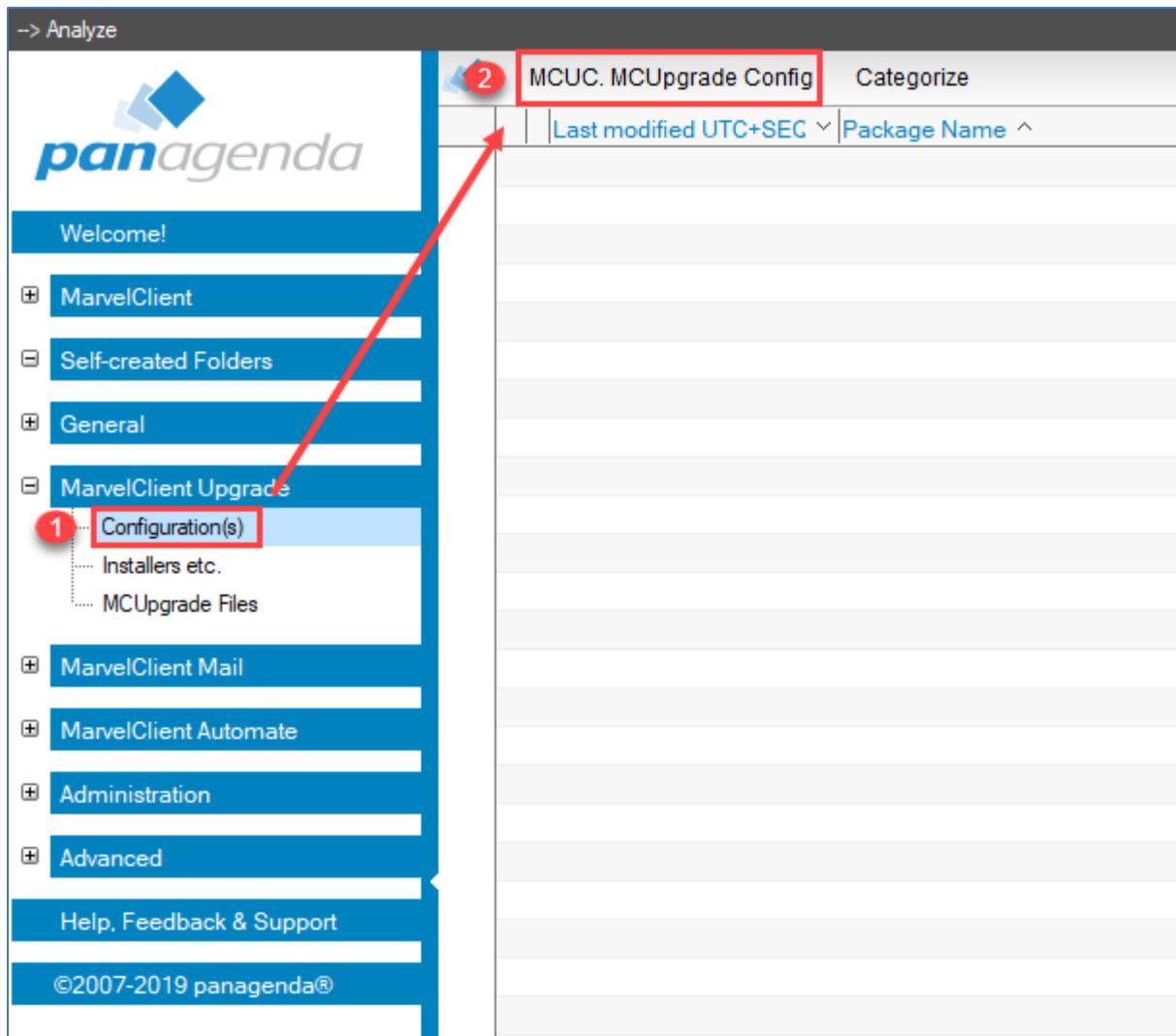


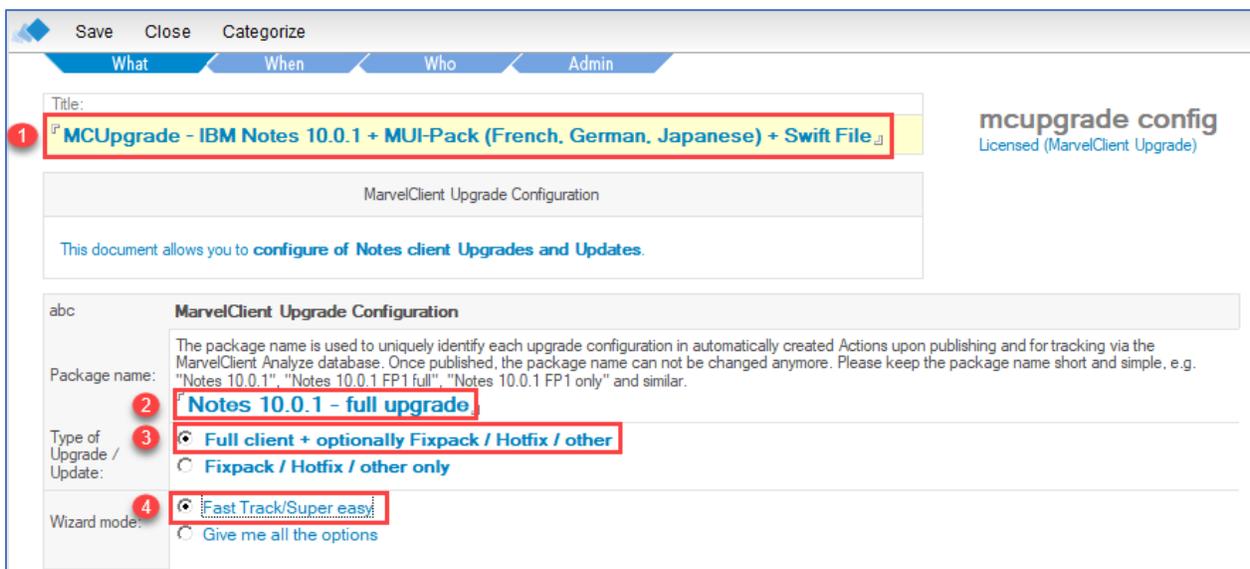
Figure 2: Create a new "Upgrade Configuration" action

In the upper part of this Action, specify the “Title” and “Package name”, for example:

1. **Title:** MCUUpgrade - IBM Notes 10.0.1 + MUI-Pack (French, German, Japanese) + Swift File
2. **Package name:** Notes 10.0.1 - full upgrade

Additionally, options for “Type of Upgrade / Update” and “Wizard mode” must be selected. For this scenario we choose

3. **Type of Upgrade / Update:** Full client + optionally Fixpack / Hotfix / other
4. **Wizard mode:** Fast Track / Super easy



The screenshot shows the 'mcupgrade config' window with the following configuration details:

- Title:** MCUUpgrade - IBM Notes 10.0.1 + MUI-Pack (French, German, Japanese) + Swift File
- Package name:** Notes 10.0.1 - full upgrade
- Type of Upgrade / Update:** Full client + optionally Fixpack / Hotfix / other
- Wizard mode:** Fast Track/Super easy

Figure 3: General part of "Upgrade Configuration" action

Note:

To use all options for your upgrade, select “Give me all the options”. We will not use this option for our herein described scenario.

2.4. Distribution tab

On the Distribution tab you can choose whether the package distribution to your clients will be handled by MarvelClient itself or your very own solution. If MarvelClient is to deploy the package, you must specify from which source the clients will download the upgrade package. You can choose between a file path (UNC or a mapped drive) and a URL (https or http).

Note:

If you need multiple download locations (for different geographies or sites, for example), you can later duplicate the so called “Smart File Downloader” Actions after publishing your upgrade configuration.

On the Distribution tab you can also index your installation packages to select them for your Upgrade.

In our current scenario we will use MarvelClient for the package distribution, using the following UNC path → [\\fileservr.local\MCUpgrade\\$](\\fileservr.local\MCUpgrade$)

Note:

This path will not work in your environment, so please choose one which exists in your environment.

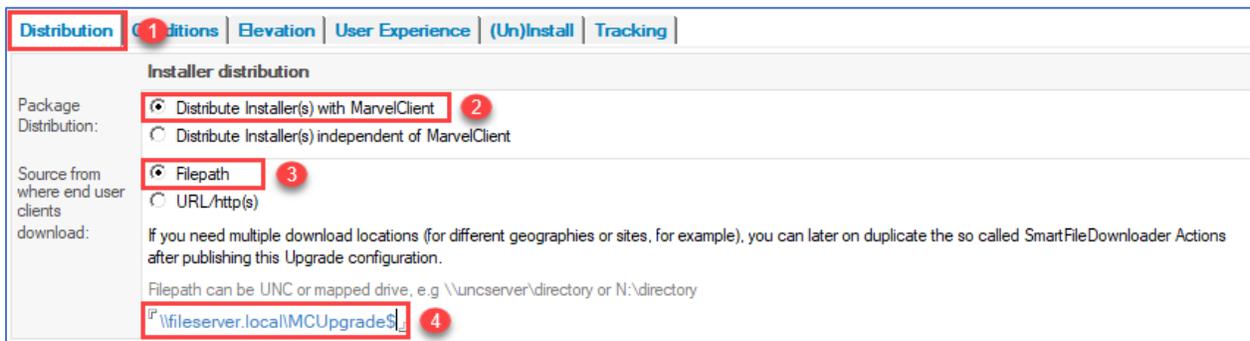


Figure 4: Upgrade Package Distribution

To index the “Full Client Installer”, click on “Index Installer(s)”, select the IBM Notes 10.0.1 installation package and click on “Open”.

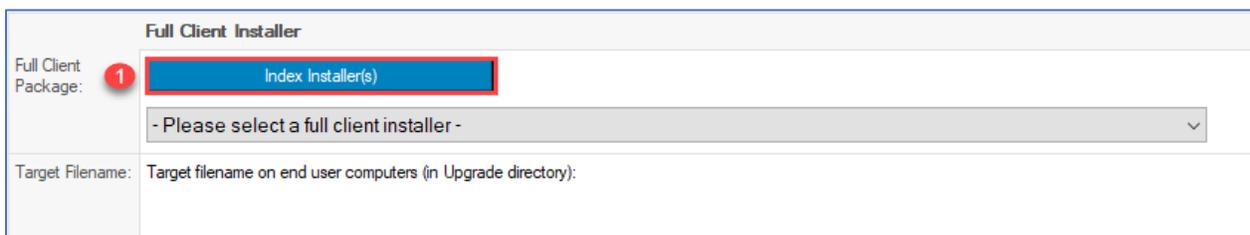


Figure 5: Import Full Client Installer 1/4

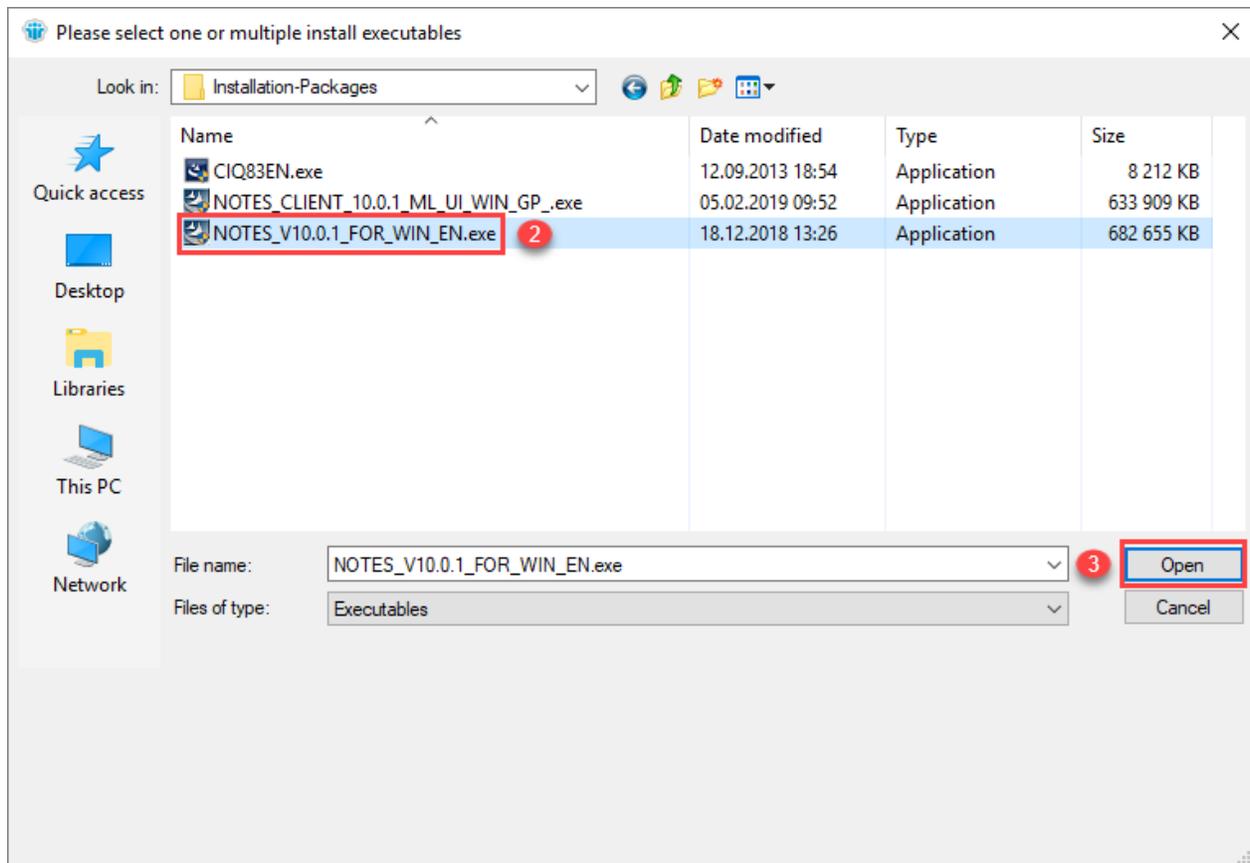


Figure 6: Import Full Client Installer 2/4

MarvelClient will then automatically create the working folder for the build process (“%public%\MCUpgrade\build), detach MCUpgrade.exe and license_upgrade.xml into that folder and use these files to

- scan the installation package,
- create a SHA256 hash of it,
- and import the meta-information into a newly created installer document in the MarvelClient configuration database.



Figure 7: Import Full Client Installer 3/4

You can now select the imported “Full Client Installer” by using the drop-down list:

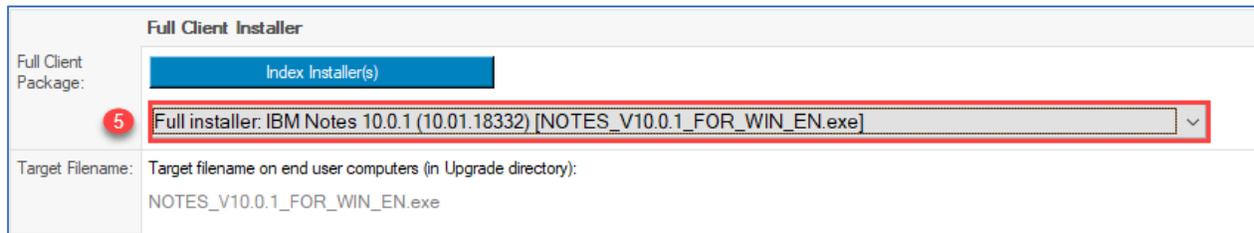


Figure 8: Import Full Client Installer 4/4

Similar to indexing and selecting the full client install package, we can now also add the MUI-Pack Group 1 and the IBM Swift File installer packages:

Simply click on “Index Installer(s)” in the “Addon / Fixpack / Hotfix / Language Pack(s) / other Installation installer(s)” section and select both installation packages:



Figure 9: Import additional Installers 1/4

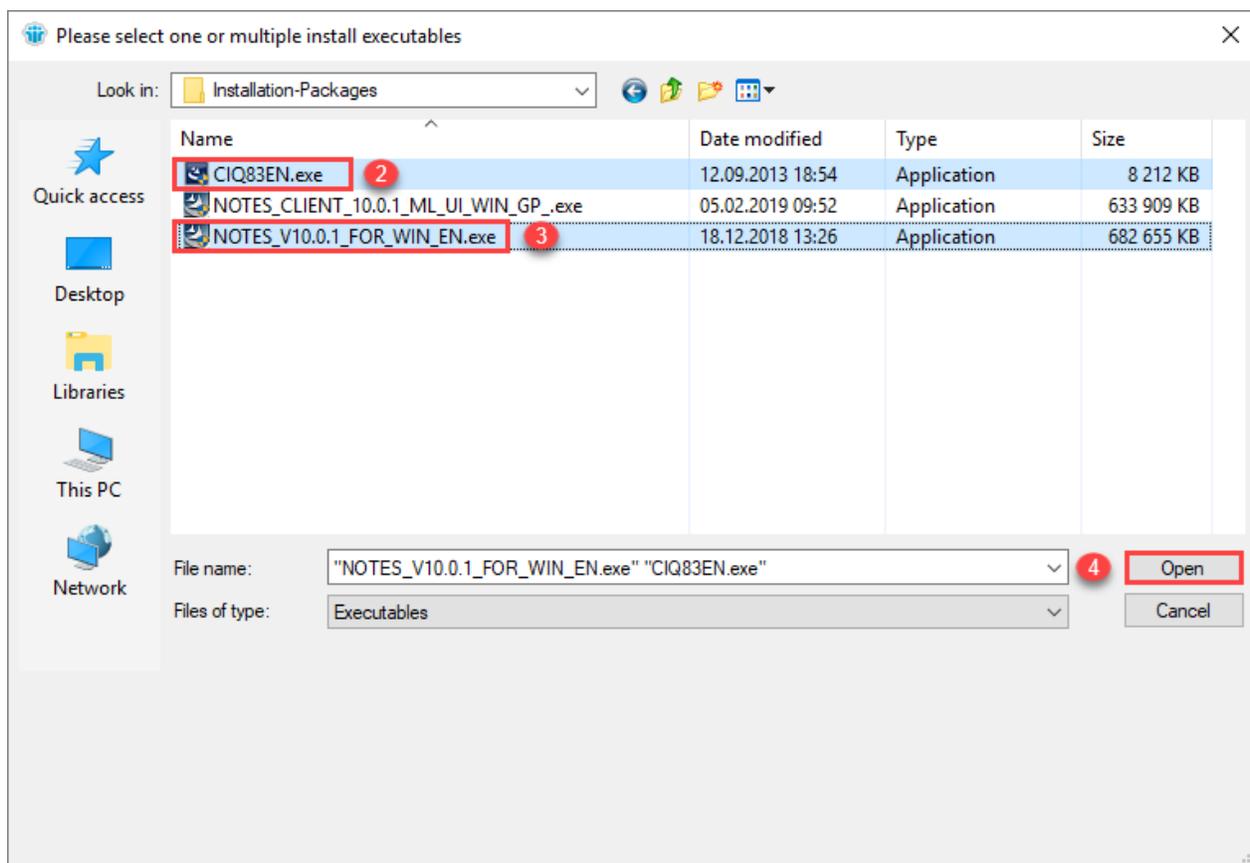


Figure 10: Import additional Installers 2/4

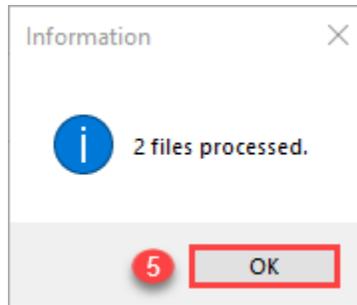


Figure 11: Import additional Installers 3/4

After the import process has finished successfully, we then select both installer packages in the checkbox field:

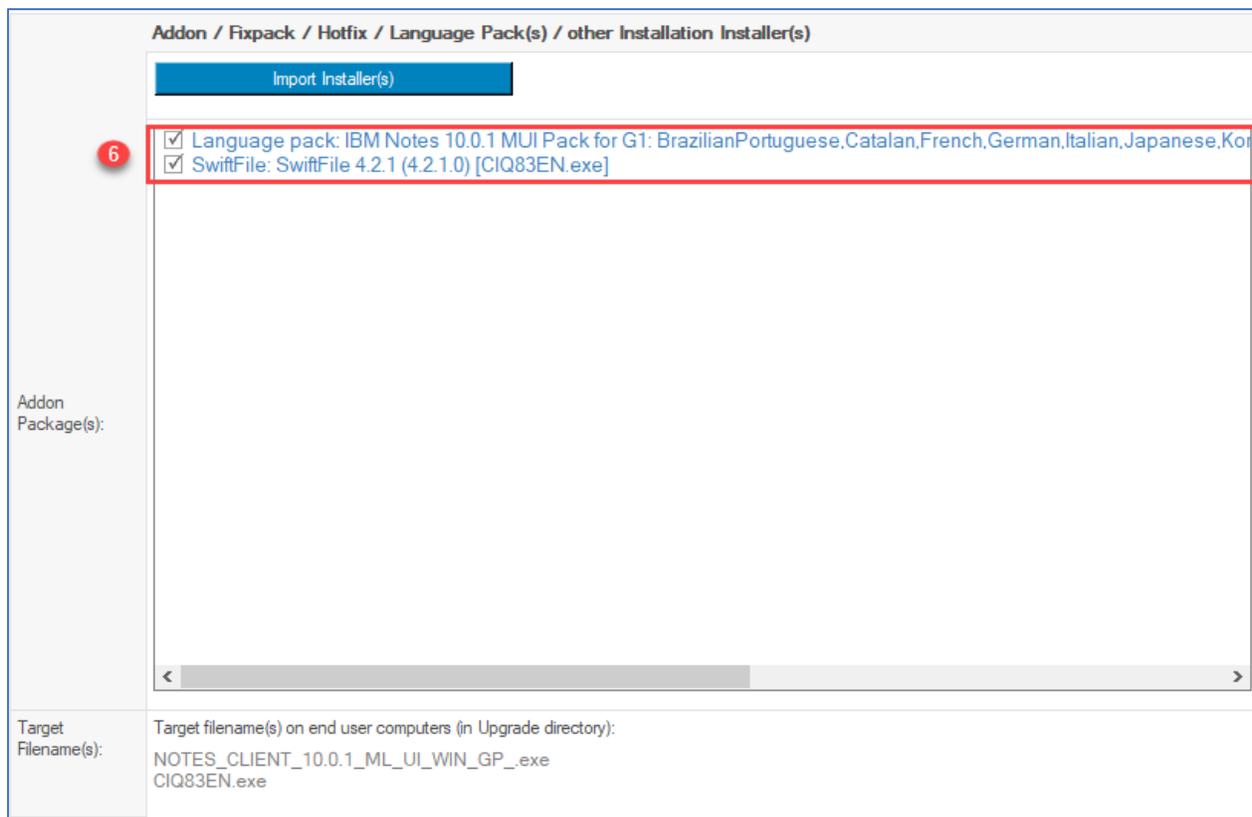


Figure 12: Import additional Installers 4/4

Tip 1:

This is a good moment to save your “Upgrade Configuration” Action, if not done yet.

Tip 2:

Please make sure, that all selected installation packages are copied to [\\fileserv.local\MCUpgrade\\$](\\fileserv.local\MCUpgrade$) (as specified in 2.4.) and that all users can access it and have read-rights to it.

2.5. Conditions tab

One of MarvelClients' many strengths is the possibility to run Actions based on Conditions. On the Conditions tab you can easily choose the required conditions for our upgrade configuration. To keep it simple, here is a list of the conditions, we will use in this scenario:

- Only for **non**-Admin and/or -Designer clients
- Only for clients for which the Notes Data Folder is **not** on a network drive
- Only if the current Notes Client version is smaller than 10.0.1
- Only on Windows 10 computers
- Only on **non**-Citrix clients

Distribution	Conditions 1	Activation	User Experience	(Un)Install	Tracking
Simple Conditions (for more Conditions, see When tab at the top)					
Admin/Designer:	<input checked="" type="radio"/> Exclude clients with Admin and/or Designer client 2 <input type="radio"/> Include clients with Admin and/or Designer client <input type="radio"/> Only upgrade clients with Admin and/or Designer client				
Data on network drive:	<input checked="" type="checkbox"/> Do not run Upgrade if the Notes data directory is on a network drive 3				
Target Notes version:	<input checked="" type="checkbox"/> Run Upgrade only if client is older than the following release ... 4 <input type="text" value="10000100"/> <small>Examples: 10000100 for 10.0.1, 9000109 for Notes 9.0.TFP9; also see notes:version_number</small>				
Target Hotfix:	<input type="checkbox"/> Run Upgrade only if client release string does not contain ...				
Target OS version:	<input checked="" type="checkbox"/> Run Upgrade only if windows build is at least ... 5 <input type="text" value="14393"/> <small>Examples: Windows 7.0= 7600, 7.0SP1=7601, 8.0=9200, 8.1=9600, 10.0 >= 14393, 15063, 16299, 17134, ... Also see os:version_build</small>				
Exclude Citrix:	<input checked="" type="checkbox"/> Citrix Clients are always excluded. 6				

Figure 13: Conditions tab

2.6. Elevation tab

Most users (>95%) do not have Windows administrator rights. To be able to install software on a user's computer, MarvelClient allows you to choose from the following options:

- Run via MarvelClient → using a local or domain account with local administration rights
- Run via MarvelClient → using the MarvelClient Upgrade Windows Service
- Deploy and configure the upgrade with MarvelClient and run via your own solution
- Static configuration and deploy and run via your own solution

The different options are described in detail in chapter [3. MarvelClient Upgrade – Advanced](#)

- [3.2 Authentication method: Local Built-In Admin](#)
- [3.3 Authentication method: Local or domain users](#)
- [3.4 Authentication method: Windows Service](#)

In our current scenario, we will use “Run via MarvelClient → MCUUpgrade as Admin” = “using a local built-in administrator account” (2). To ensure that the Upgrade is exclusively done with your provided Admin user credentials, we also check both “Do not run Upgrade ...” options (3).

Please follow the steps in chapter [3. MarvelClient Upgrade – Advanced](#) to create the needed keyfile with administrative credentials (.mcu file) and make sure it is then attached it to the last field on this tab (4).



Figure 14: Elevation tab using local built-in administrator account

2.7. User Experience tab

During the Upgrade process, MarvelClient Upgrade will show a Splash Screen with progress bars. MarvelClient Upgrade Free customers can choose between 10 different designs, which can be configured on the User Experience Tab. A custom Splash Screen can only be used by MarvelClient Upgrade licensed customers.

For our scenario we leave the default Splash Screen #3 (2) with yellow font (3).

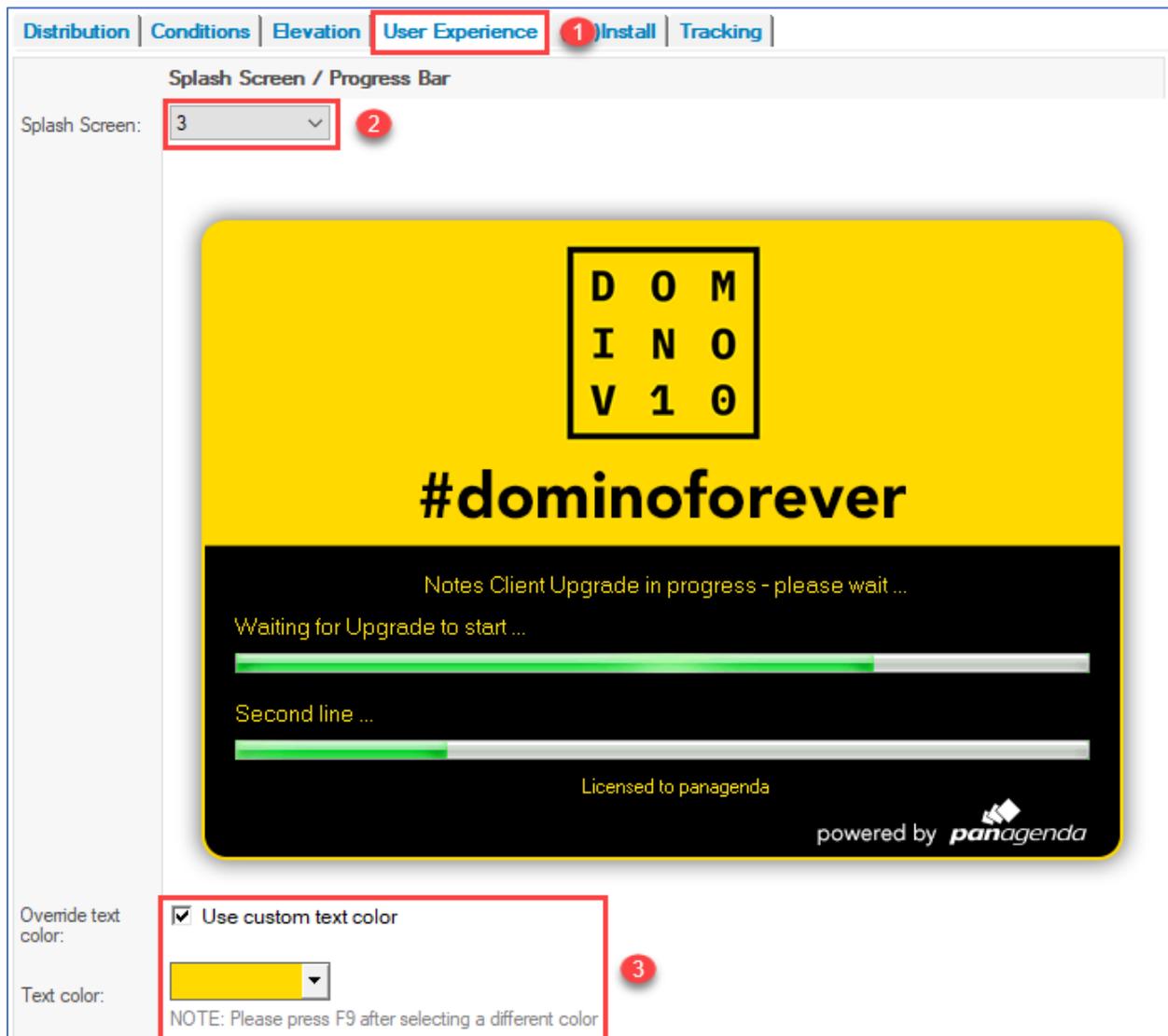


Figure 15: User Experience tab

2.8. (Un)Install tab

One of the most important tabs of the MarvelClient Upgrade Wizard is the (Un)Install tab, which controls

- whether the current client will be uninstalled before the new installation gets executed,
- and the exact command lines for the full client and/or additional install packages and their order of execution.

If you are not just installing a Fix Pack and/or other additional packages, but a full client package, then we highly recommend uninstalling the current IBM Notes client using [NICE](#) (2). Compared to an in-place upgrade, the success rate of installing a Notes client is **much** higher after uninstalling the previous Notes client. Don't be scared, the Data folder will **not** be touched when using this option.

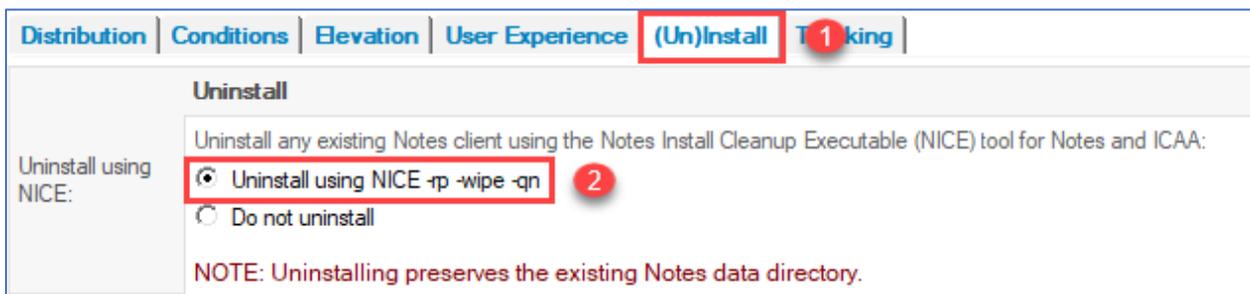


Figure 16: Uninstall

Based on the meta information of the selected packages on the Distribution tab, the MarvelClient Upgrade Wizard can automatically generate the best possible command line to install your package(s): Simply click on “Fill below fields according to Distribution tab” (Fig. 17 (1) and Fig. 18 (1)). The %MC_UPGRADEDIRECTORY% variable in the resulting command line reflects the working folder of MarvelClient Upgrade, which defaults to “%PUBLIC%\MCUpgrade”.

Note:

Unless you choose for a static splash screen progress text, the content of the InstallText field will be shown during the upgrade process on the Splash Screen.

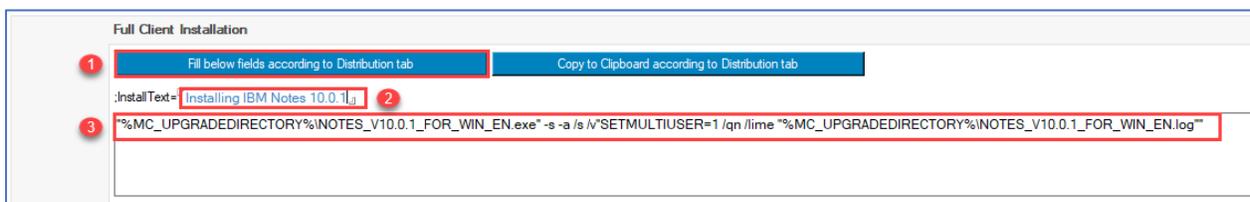


Figure 17: Full Client Installation command line

Next we also click on the “Fill below fields according to Distribution tab” button (1) in the add-on installation section to fill in the InstallText and InstallExec instructions for the MUI-Pack and IBM Swift file:

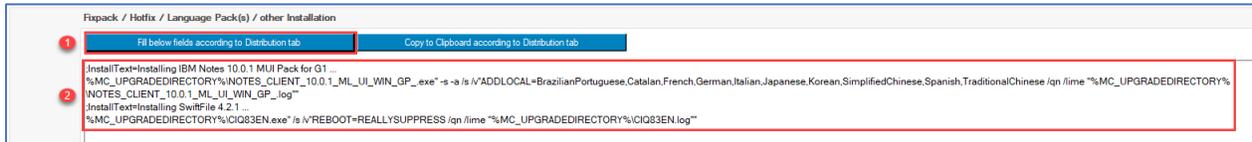


Figure 18: "Fixpack / Hotfix / Language Pack(s) / other Installation" command lines 1/2

After doing so, we need to make some changes to both the InstallText and InstallExec command line for the MUI-Pack installation, so that only French, German and Japanese are installed instead of all languages included with the selected language pack:



Figure 19: "Fixpack / Hotfix / Language Pack(s) / other Installation" command lines 2/2

Note:

When using the word “Installing “ at the beginning of an InstallText line, it will later on be automatically translated for progress display.

2.9. Tracking tab

MarvelClient Upgrade is able to send so called “interim progress messages” during an upgrade, as well as final “success” or “fail” reports at the end of an upgrade via SMTP.

The recommended target for such SMTP emails is the MarvelClient Analyze database, as it aggregates all progress into one single document per user and upgrade package.

Note:

The MarvelClient Analyze database then needs to be a Mail-In database with an external email address.

On the Tracking tab you can fill in the necessary SMTP configuration. Should you require SMTP authentication, please see chapter [3. MarvelClient Upgrade – Advanced](#)

→ `i_MCUpgrade.mcu`

... to provide authentication credentials through the encrypted MCUUpgrade keyfile (.mcu).

In our current scenario we will use the following configuration (which will of course **not** work in your environment, so please use your own configuration here):

- SMTP-Server: smtp.panagenda.local
- From Name: %USERNAME% on %COMPUTERNAME%
- From and To (Email) Address: yourMCAalyzeMailInName@yourdomain.com
- Use SSL: Yes, SMTP Port: 0
(if the port is set to 0, it defaults to port 25 without SLL and 465 with SSL)

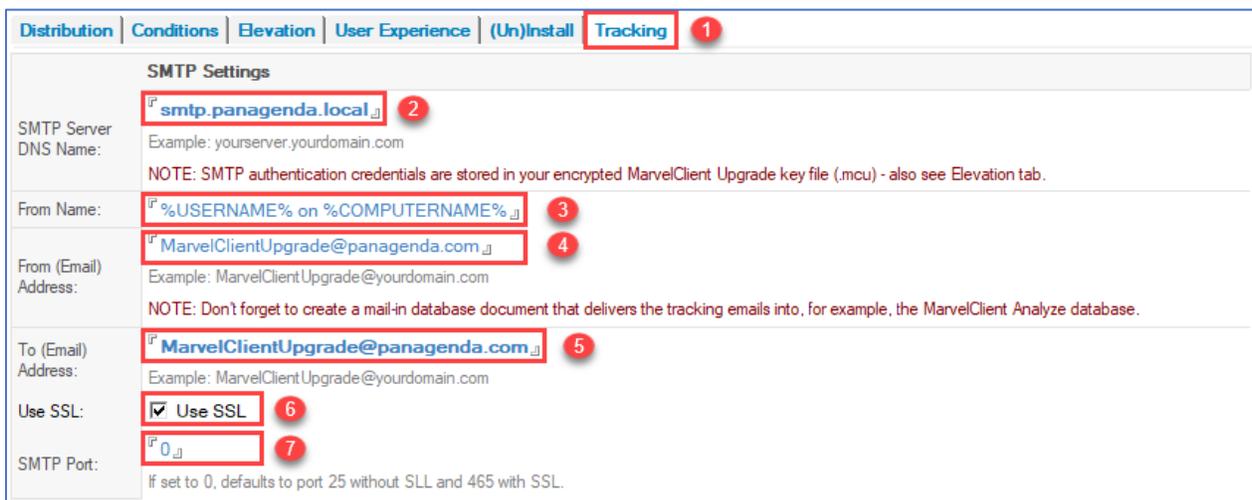


Figure 20: Tracking tab (SMTP configuration)

2.10. Prepare or Publish Upgrade

The almost very last thing we need to do now is to publish our “Upgrade Configuration”.

IMPORTANT: Before doing so, you may want to further restrict your upgrade to a test user or group. To do so, please click on the “Who” tab at the very top of your Upgrade Configuration document, select “ONLY FOR” and enter the respective user and/or group name(s) into the direct input field.

We recommend that you also check the EXCLUDE option (you do not have to enter any users or groups into the direct input field in the EXCLUDE section) to make sure that all users with role [Admin] in the MC Configuration database are excluded, too.

The “Prepare or Publish Upgrade” button will first prepare all necessary components in %PUBLIC%\MCUpgrade\stage. Upon completion of preparation, it will ask you whether you want to publish the upgrade into MarvelClient Action that will take care of the rest :)

Customers who want to use MarvelClient Upgrade with their very own software deployment solution can simply choose not to publish and use the preparation results from the folder %PUBLIC%\MCUpgrade\stage.

In our scenario, we choose yes for publishing, which then creates all needed MarvelClient Actions automatically, so that all corresponding Notes client are automatically upgraded to Notes 10.0.1.

Note:

Once again, please make sure to configure your test scope and respective target users on the “When” and the “Who” tab before going for “all users at once”:

On the “When” tab you can manage additional conditions for the resulting upgrade actions.

On the “Who” tab, you can specify who will execute (ONLY FOR) or not execute (EXCLUDE) the client upgrade.

If you do not configure any of the above tabs, then ALL USERS will get the client upgrade.

More information about the When and Who tab can be found here:

<https://www.panagenda.com/kbase/display/mc/The+What%2C+When%2C+Who+and+Admin+Tabs>

What	When	Who	Admin
Title: MCUpgrade - IBM Notes 10.0.1 + MUI-Pack (French, German, Japanese) + Swift File			mcupgrade config Licensed (MarvelClient Upgrade)
MarvelClient Upgrade Configuration			
This document allows you to configure of Notes client Upgrades and Updates .			
MarvelClient Upgrade Configuration The package name is used to uniquely identify each upgrade configuration in automatically created Actions upon publishing and for tracking via the MarvelClient Analyze database. Once published, the package name must not be changed anymore. Please keep the package name short and simple, e.g. "Notes 10.0.1", "Notes 10.0.1 FP1 full", "Notes 10.0.1 FP1 only" and similar.			
Package name:	<input type="text" value="Notes 10.0.1 - full upgrade"/>		
Type of Upgrade / Update:	<input checked="" type="radio"/> Full client + optionally Fixpack / Hotfix / other <input type="radio"/> Fixpack / Hotfix / other only		
Wizard mode:	<input checked="" type="radio"/> Fast Track/Super easy <input type="radio"/> Give me all the options		
Click on the following button when you are done with reviewing / filling in the below table:			
<input type="button" value="Prepare or Publish Upgrade"/> 1			

Figure 21: Prepare or Publish Upgrade 1/3

Publish Upgrade ✕

? You are creating / updating the MarvelClient Upgrade actions.
 The actions are categorized as follows:
 >> MCUpgrade\Notes_10.0.1_FP1_NEW <<

Should the actions be enabled?

2

Figure 22: Prepare or Publish Upgrade 2/3

Publish Upgrade ✕

i **Publish Upgrade FINISHED: 7 Action(s) created / 1 Action(s) updated / 7 Action(s) enabled!**

3

Figure 23: Prepare or Publish Upgrade 3/3

The resulting Actions (5) will be shown in the Rich Client Essentials / All Actions (4) views:

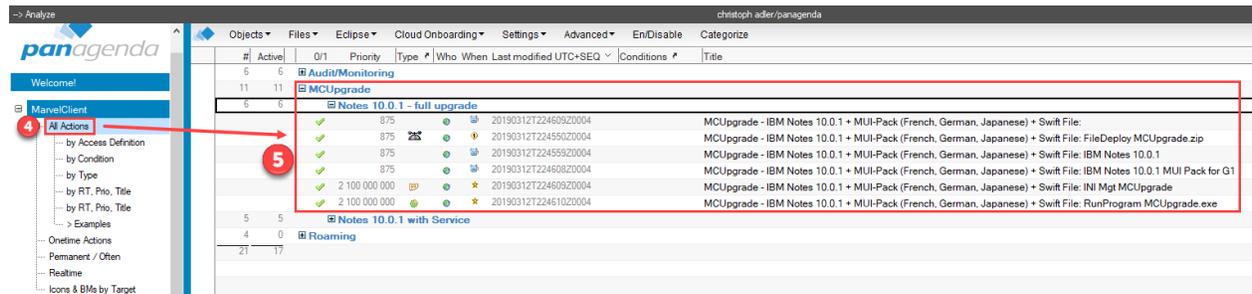


Figure 24: MarvelClient Upgrade Actions

If you decide to later on make changes to your “Upgrade Configuration” and then click on “Publish” again, all changes will be applied to your previously published upgrade Actions automatically.

3. MarvelClient Upgrade - Advanced

Normal end users usually do not have enough permissions to install software on their computers. With MarvelClient Upgrade you have multiple options to solve this challenge:

You can use the following options to provide MarvelClient Upgrade with increased rights:

1. Switch to the Local Built-In Administrator
2. Switch to a domain administrator or a domain-wide installation user
3. Use the MarvelClient Upgrade Service

If needed, you can also provide MarvelClient Upgrade with SMTP authentication credentials for sending upgrade intermediate and final status messages and reports.

The following pages describe the components of MarvelClient Upgrade needed for implementing one or multiple of the above options.

System requirements: .NET 3.5 or .NET 4.x.

3.1. Components

The following tools and components are used to provide elevated user rights:

- MCUUpgrade.mcu – the encrypted MarvelClient Upgrade keyfile
- Service/Whitelisting only: MCKeyfileEditor.exe
- Service/Whitelisting only: Whitelisting Editor
- Service/Whitelisting only: MCUUpgrade.ini

Not all components are used equally in every authorization method. However, this section is intended to give you an overview of all the components.

i. MCUUpgrade.mcu

The "MCUpgrade.mcu" is a key/password store. It is created using MCKeyfileEditor, required for each of the above authentication methods, and is included in a prepared/published upgrade. When using the MCUUpgrade Service, no credentials are stored in the keyfile, except for SMTP.

ii. MCKeyfileEditor.exe

With the help of "MCKeyfileEditor" you generate the above mentioned "MCUpgrade.mcu". You can also use this tool to edit and complete an existing "mcu" file, if necessary. In addition, you can also use MCKeyfileEditor to access the "Whitelisting Editor" which is required for using the Upgrade service.

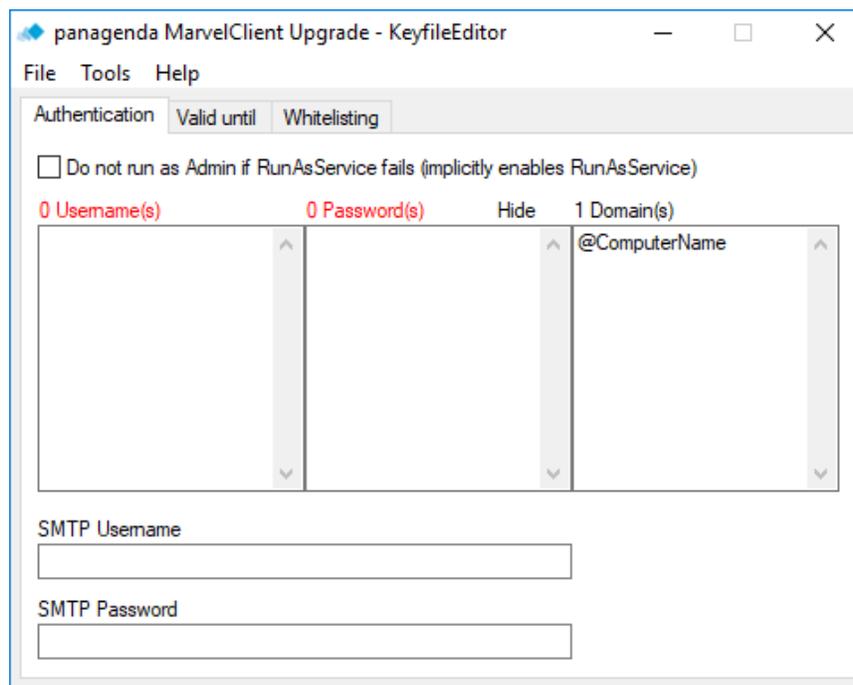


Figure 25: The MCKeyfileEditor

iii. Whitelisting Editor

The "Whitelisting Editor" can be accessed from within "MCKeyfileEditor" and is used in combination with the MCUUpgrade Service.

It is used to ensure that your upgrade package cannot be exploited, by hashing all zip files and executables, and signing (un)install and other instructions with your very own public and private keypair.

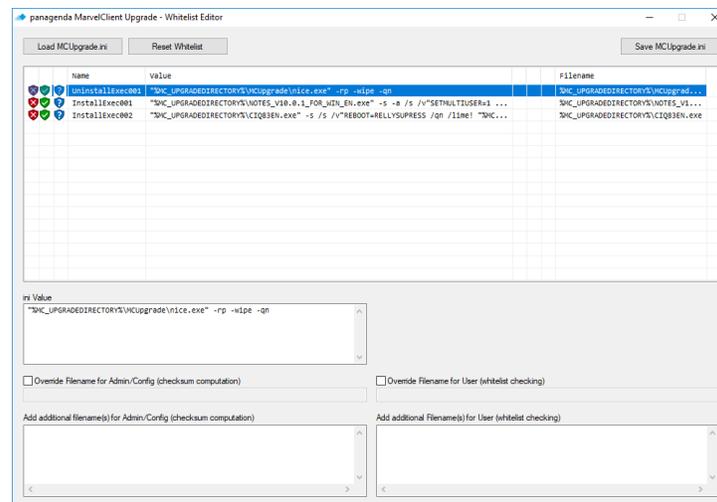


Figure 26: The Whitelisting Editor

To a large extent, the whitelisting process is automated when preparing/publishing your upgrade. It is further detailed out in chapter 3.b.iii on page 26.

iv. MCUUpgrade.ini

"MCUpgrade.ini" contains your upgrade configuration and controls the "MarvelClient Upgrade" process. It is generated and completed using the information you entered in the MarvelClient Config database. Each upgrade package includes its own "MCUpgrade.ini".

v. MarvelClient Upgrade Build Directory

The MarvelClient Upgrade Build Directory can be found in the following path:

- %public%\MCUpgrade\build

3.2. Authentication method: Local Built-In Admin

If you use the same password for the Windows Local Built-In Admin across all your clients, this authentication method is an easy way to provide increased rights to MarvelClient Upgrade. Also, the User Account Control (UAC) rules do not apply to this account, which simplifies further configuration.

Note:

Because of Windows UAC it is important to not use a local user with administrator rights here, but the so called local built-in admin account. Otherwise you must follow instructions in chapter 3.3.

Name	Full Name	Description
Administrator		Built-in account for administering the computer/domain
DefaultAccount		A user account managed by the system.
Guest		Built-in account for guest access to the computer/domain

Figure 27: The Local Built-In Admin Account

1. Launch MCKeyfileEditor via the "Elevation" tab in your MarvelClient upgrade configuration document ...

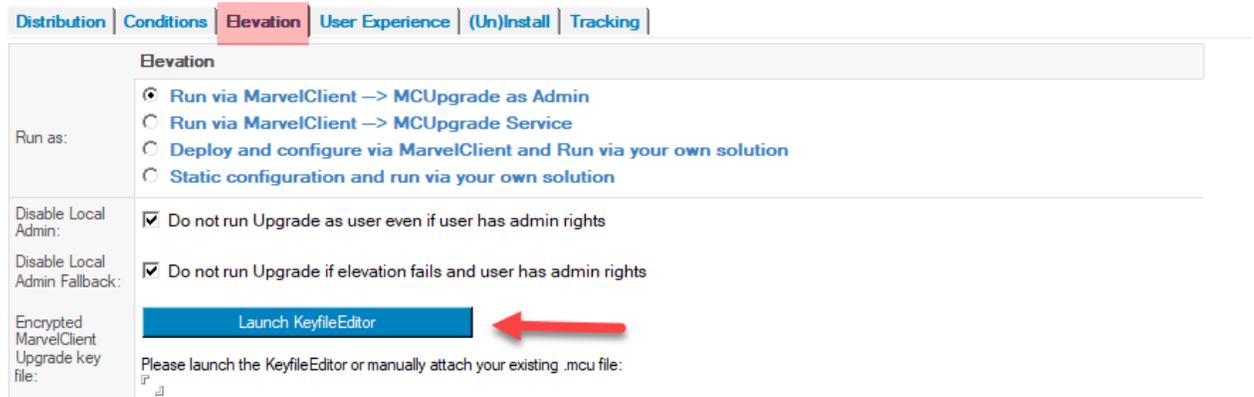


Figure 28: Launching MCKeyfileEditor

... and enter the account name and corresponding password on the following screen:

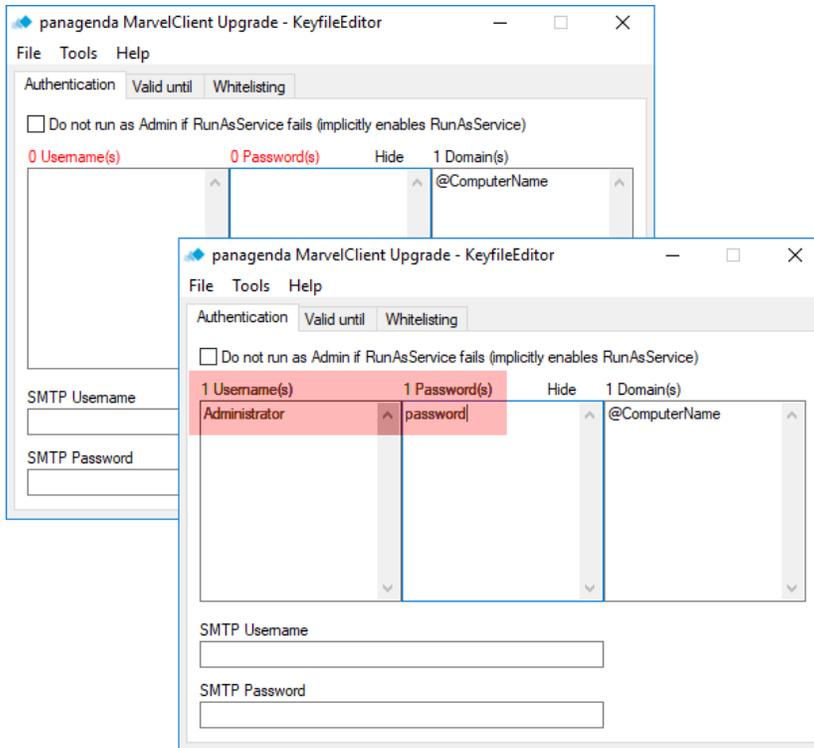


Figure 29: KeyfileEditor Admin Credential Input

For all locally used admin accounts, leave the entry "@ComputerName" under "Domain(s)" unchanged.

2. Save the "MCUpgrade.mcu" file as follows:

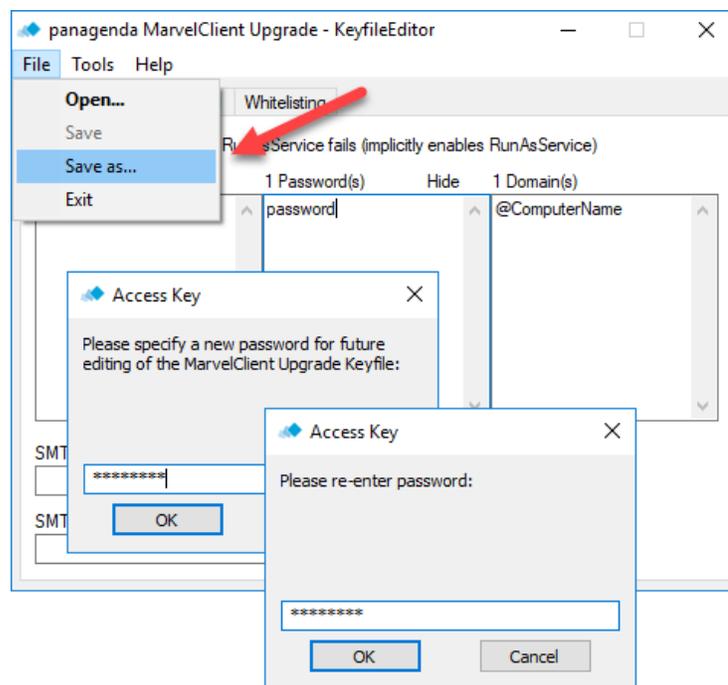


Figure 30: Saving your MCUpgrade.mcu file 1/2

Note:

The password entered during this step is used to protect the MCUUpgrade.mcu file from unauthorized viewing or editing.

Next, store your MCUUpgrade.mcu (keyfile) in the MarvelClient Upgrade Build Directory:

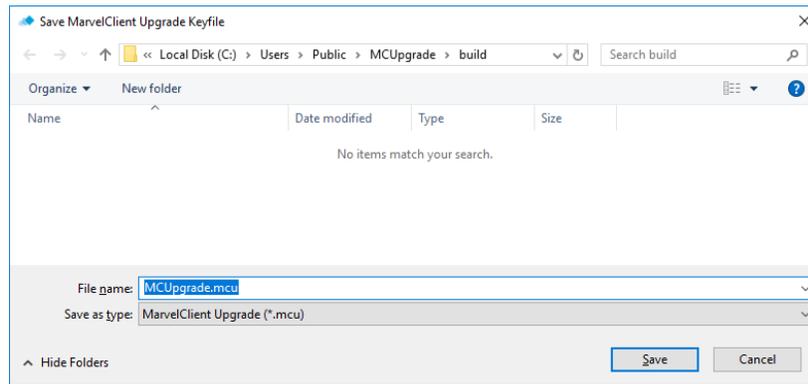


Figure 31: Saving your MCUUpgrade.mcu file 2/2

After saving your MCUUpgrade.mcu file and closing MCKeyfileEditor, your .mcu file will be attached to your upgrade configuration automatically:

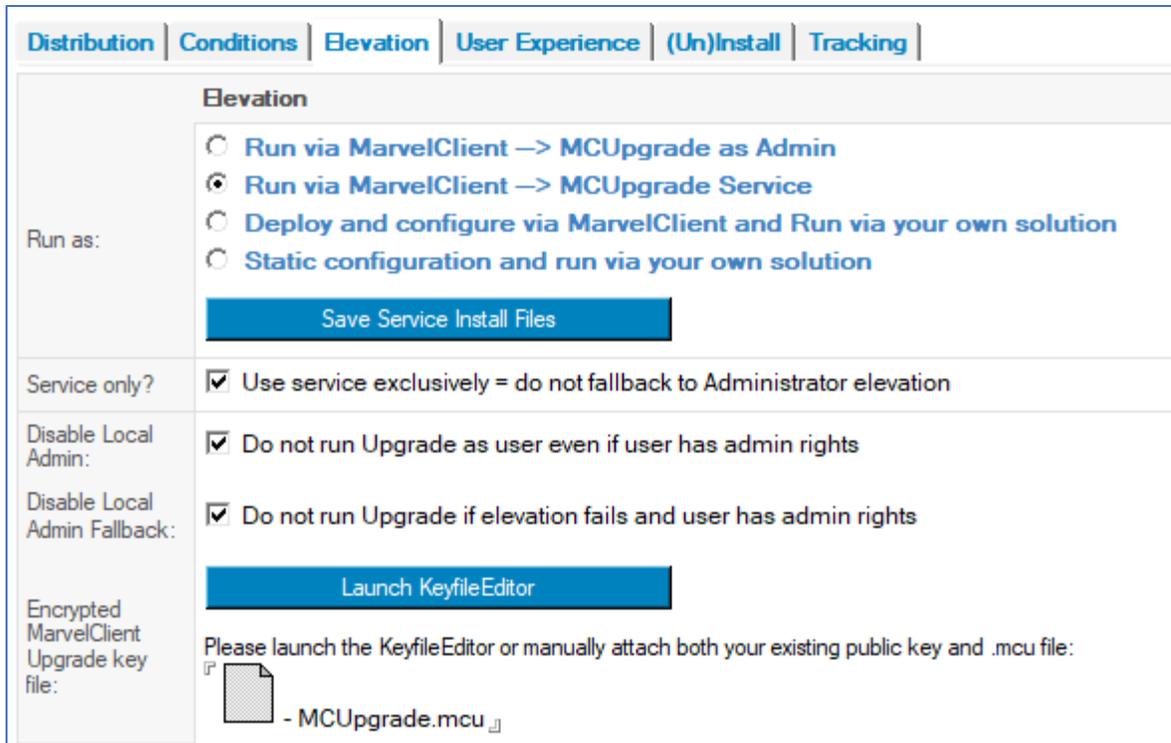


Figure 32: Attaching your MCUUpgrade.mcu file to your upgrade configuration

3.3. Authentication method: Local or domain users

Of course, you can also use a user with elevated privileges from a domain for your installation. Usually the domain administrator or an "Install-User" provided especially for this purpose is used here.

Follow the exact same steps as described in 3.2., but enter the user to be used, the password **and your domain** in the appropriate fields:

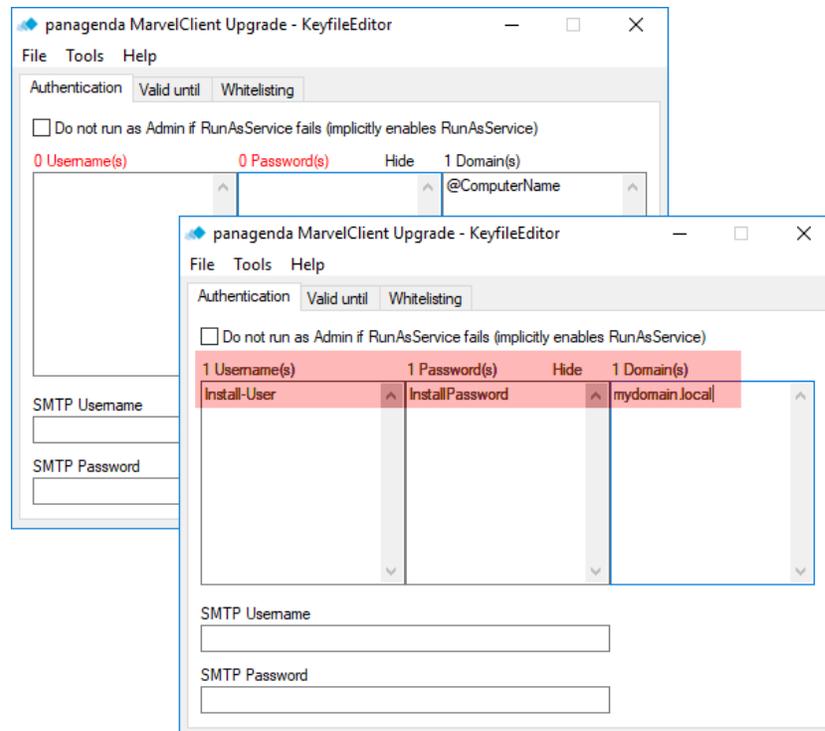


Figure 33: Providing Local or Domain user credentials

i. Use of multiple local users

It is also possible to store multiple local users in your "MCUpgrade.mcu" file. This may be necessary, for example, if local accounts with different names and passwords have been created across different locations. To do this, enter the user names and corresponding passwords one after the other in the "MCKeyfileEditor".

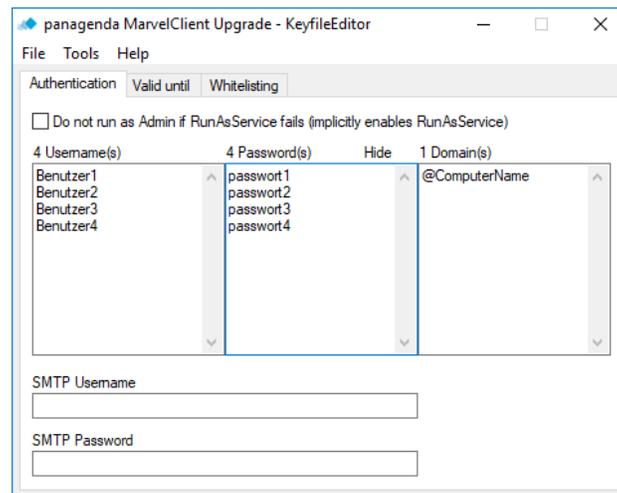


Figure 34: Using multiple usernames and passwords in MCKeyfileEditor

It is also possible to store several passwords for the same user.:

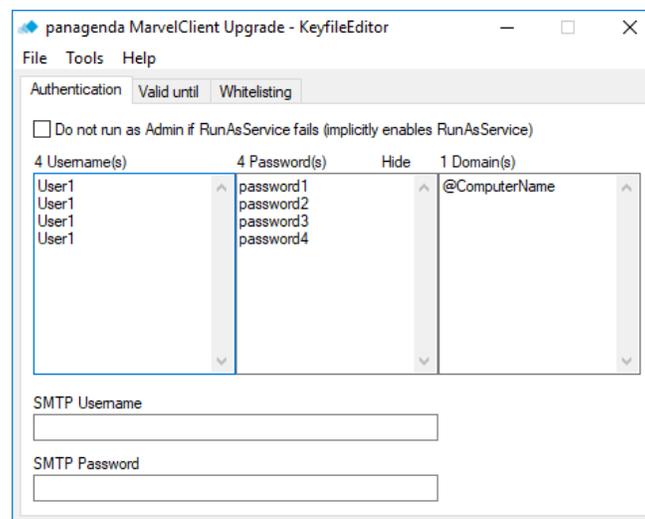


Figure 35: Using multiple passwords with the same user in MCKeyfileEditor

Note:

You may also use logon permutation, meaning an unequal number of user names and passwords – this will result in all usernames being tried in combination with all passwords. In line with this, please make sure to also read the following IMPORTANT notice.

IMPORTANT:

Please ensure that the number of maximum possible logon attempts does not lead to the used user account being blocked.

When using a local Administrator account, the maximum number of failed attempts counts as per workstation/operating system user.

When using a domain Admin account, the number of failed logon attempts add up across all workstations/operating system users!

ii. Configuration of Windows User Account Control (UAC)

As soon as a user account is used which is not the so-called "local built-in administrator" account, the mechanisms of the User Account Control (UAC) in Windows apply.

By default, an elevation request (=requiring increased rights) results in a popup window for your end users. This window asks your end users to confirm and acknowledge that a different user context will be used from this point on, even if with MarvelClient Upgrade end users are no longer asked for an administrative password.

For a variety of reasons, this behavior is hardly practicable. On the one hand, the user may not be able to perform the confirmation (when being absent, for example), and on the other hand, there it is quite likely that the user will not perform the confirmation and hence the upgrade would terminate.

To solve the situation, the user account control (UAC) needs to be adjusted as follows via Windows Group Policy Object (GPO): In the Local Security Guideline, set the value of ...

User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode

Figure 36: UAC Admin Approval Mode

... from "Prompt for consent" ...

Figure 38: UAC: Prompt for consent

to: „Elevate without prompting“:

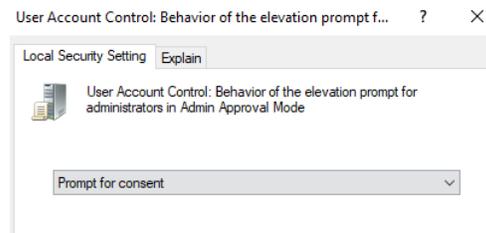


Figure 37: Admin Approval Prompt for consent

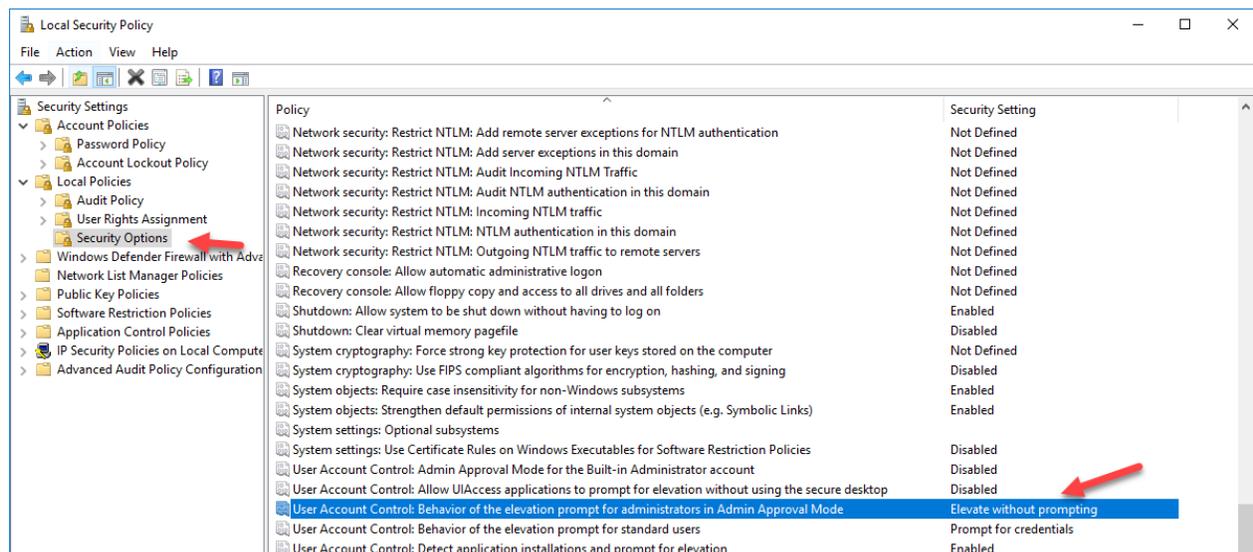


Figure 39: Local Security Policy

Alternatively, you can set the following registry values for your users:

The registry keys are found in `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System`. For information about each of the registry keys, see the associated Group Policy description.

Name	Type	Data
 (Default)	REG_SZ	(value not set)
 ConsentPromptBehaviorAdmin	REG_DWORD	0x00000000 (0)

Figure 40: ConsentPromptBehavior Registry setting

The possible settings for this key are as follows:

<p>ConsentPromptBehaviorAdmin</p>	<p>User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode</p>	<p>0 = Elevate without prompting 1 = Prompt for credentials on the secure desktop 2 = Prompt for consent on the secure desktop 3 = Prompt for credentials 4 = Prompt for consent 5 (Default) = Prompt for consent for non-Windows binaries</p>
-----------------------------------	---	---

We would like to emphasize that the above does not deactivate User Account Control, but only suppresses the display of elevation prompts for when authenticating using a local Admin account. All other functions of the User Account Control remain active.

3.4. Authentication method: Windows Service

Another possibility is the use of a Windows service provided by panagenda. This involves the creation of a private and public key pair for use of the so called "Whitelisting" functionality.

Basically, the preparation consists of three parts:

1. One-time deployment of the Windows service on the workstations of your users
2. One-time creation of a key pair
3. Whitelisting of each upgrade

Whitelisting ensures that the service cannot be mis-used and only instructions and software components that you have defined are executed.

The deployment of the Windows service is described in more detail in section [4. How to setup the panagenda install service](#).

i. Whitelisting explained

The whitelisting function is used to ensure that only files and instructions are executed as desired by you. A checksum is generated for both all relevant instructions, as well as all files that are to be unzipped or executed. The components of your upgrade package can therefore not be subsequently changed / manipulated.

Any relevant lines in "MCUpgrade.ini" or files which are part of your upgrade configuration, that are changed after whitelisting, will not be executed anymore.

This ensures that the Windows service cannot be mis-used and your upgrade package is tamper proof.

ii. Preparatory activities

No login information is required in the "MCUpgrade.mcu" file when using the MarvelClient Upgrade service. (SMTP authentication credentials may be an exception to this, as well as when using fallback authentication to administrator accounts)

In the configuration document under the tab "Elevation", select the method "MCUpgrade Service" (1) and afterwards click on "Launch MCKeyfileEditor" (2).

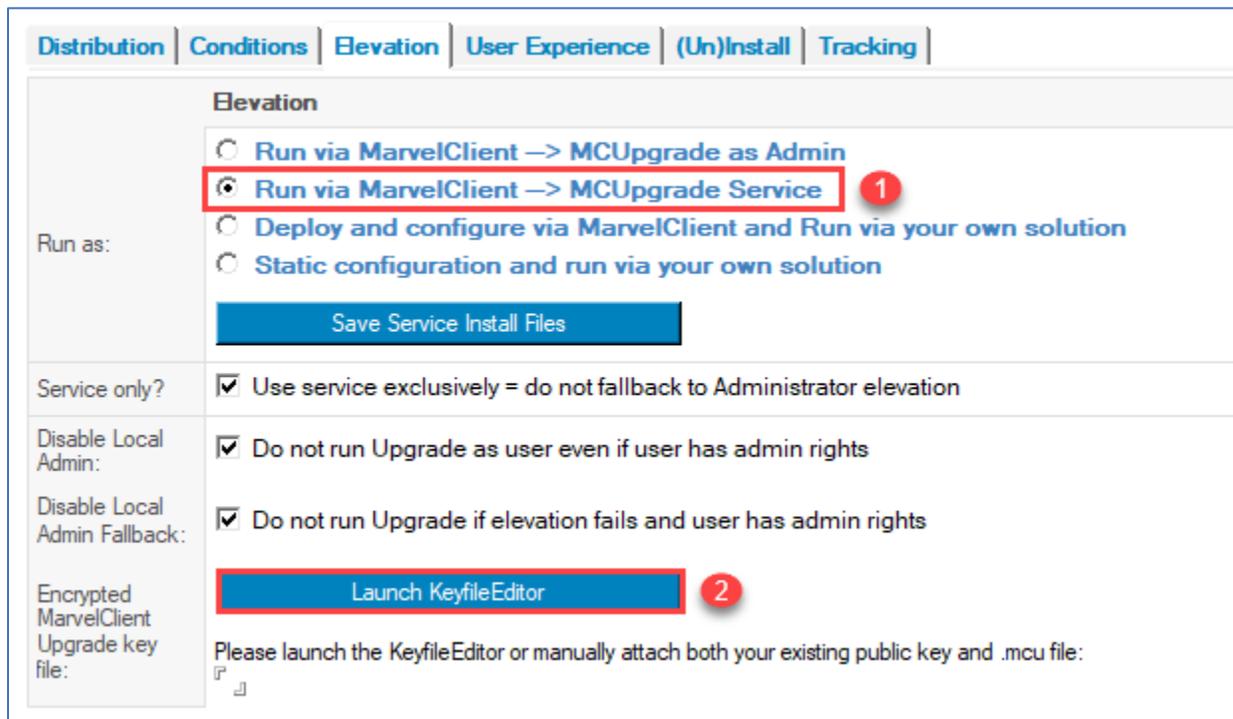


Figure 41: Launching the MCKeyfileEditor to create your keypair

In "MCKeyfileEditor" under "Tools" select the menu item "Create keypair":

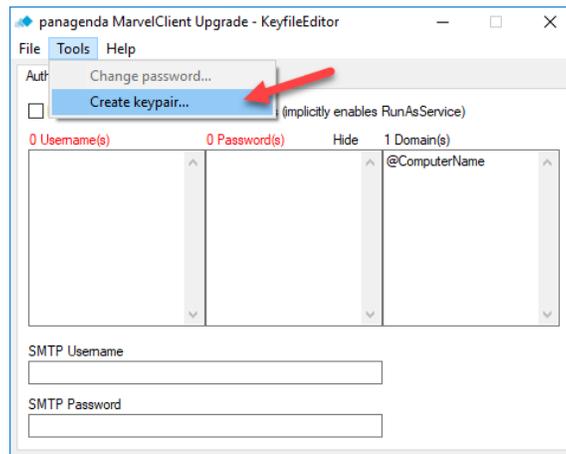


Figure 42: Create keypair menu option in KeyfileEditor

Next, your very own key pair consisting of a "private" and a "public" key is created.

Generating keypair, please wait ...

Note:

Your keypair is created in the "MarvelClient Upgrade Build Directory" = %PUBLIC%\MCUpgrade\build

-  MCUpgrade_PrivateKey_STORE_SAFELY_DO_NOT_SHARE.sig
-  MCUpgrade_PublicKey.sig

Figure 43: Your keypair filenames

Confirm the message upon completion of the key generation:

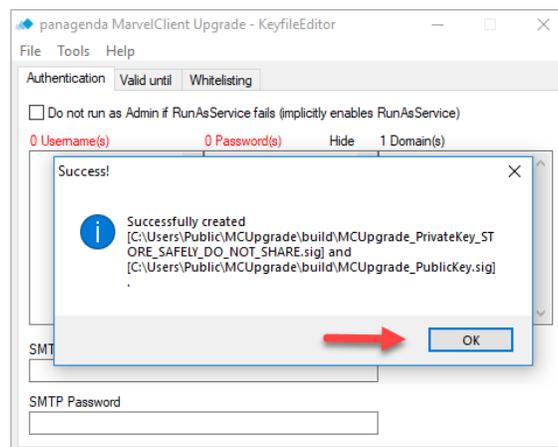
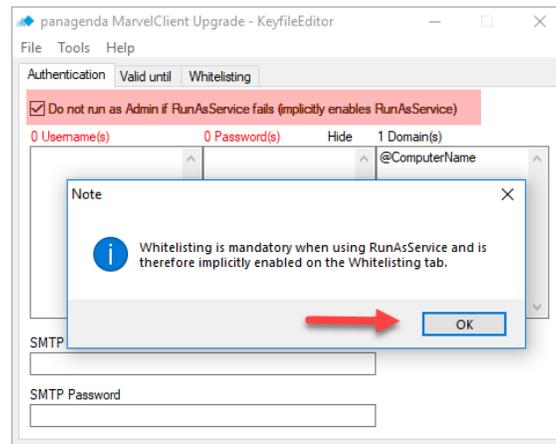


Figure 44: Message at end of keypair generation

Next, select "Do not run as Admin..." on the Authentication tab:



A message is displayed, stating that whitelisting is now implicitly enabled. Also, the input fields for entering usernames and passwords are now deactivated, because no login data is required when using the Windows service.

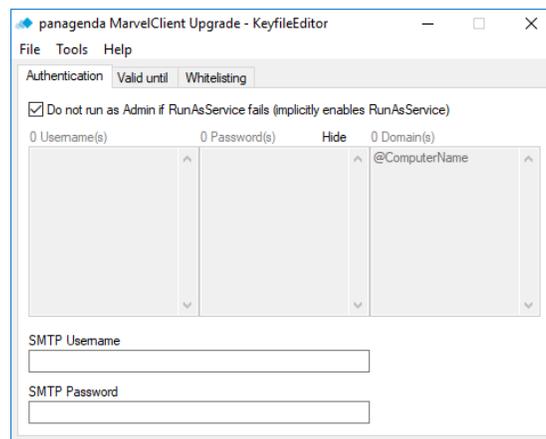


Figure 45: Exclusively using the Windows service

Note:

If necessary, you can also combine usage of the MCUUpgrade service with fallback to “logon as local built-in Administrator and/or local/Domain Admin”. In that case, do not check the box “Do not run as Admin...”, but enter one or multiple user names and passwords for authentication. This feature may be useful in environments where the service cannot be brought out on all machines.

Next, save the "MCUpgrade.mcu" file.

Note:

You may receive a warning message that “Whitelisting has been enabled but no whitelisting has taken place yet”, which you can safely ignore for now:



Figure 46: Missing Whitelist warning

Confirm this message to proceed with saving your MCUpgrade.mcu file:

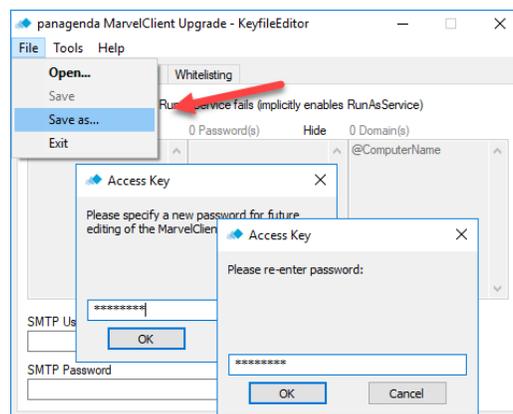


Figure 47: Securing your MCUpgrade.mcu keyfile with a password

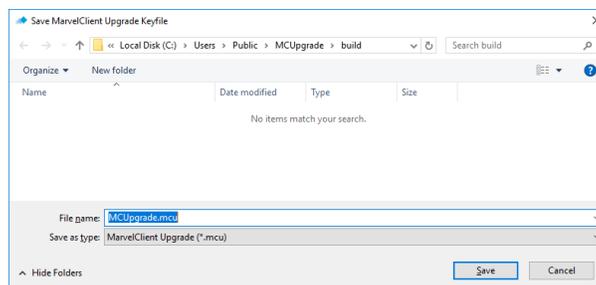


Figure 48: Saving your MCUpgrade.mcu keyfile

Now close "MCKeyfileEditor". You will notice that both the "MCUpgrade.mcu" and "MCUpgrade_PublicKey.sig" file are attached to your upgrade configuration document:

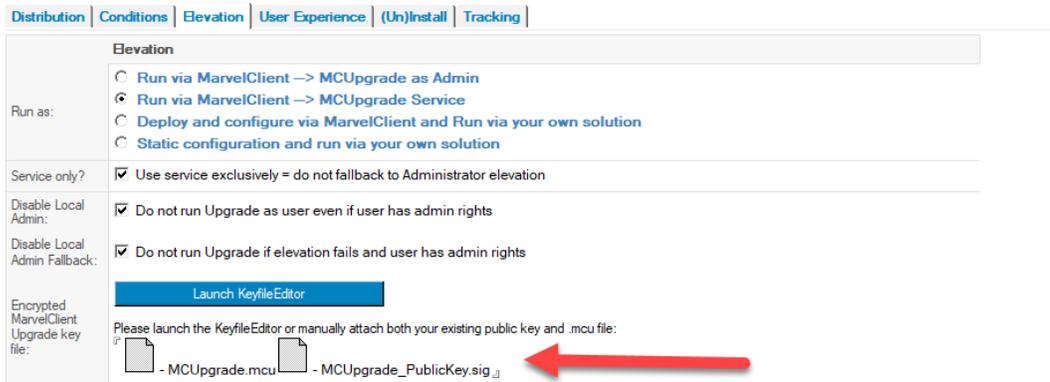
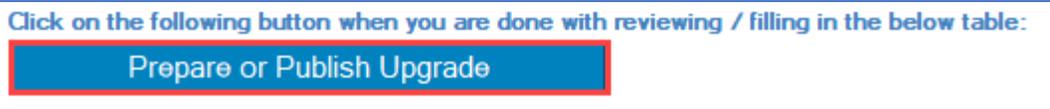


Figure 49: MCUpgrade.mcu and MCUpgrade_PublicKey.sig attachments

iii. Performing Whitelisting

As soon as you have completed your upgrade configuration in the MarvelClient configuration database and prepare/publish the package ...



... the "Whitelisting Editor" opens (only when configuring your upgrade to use the service):

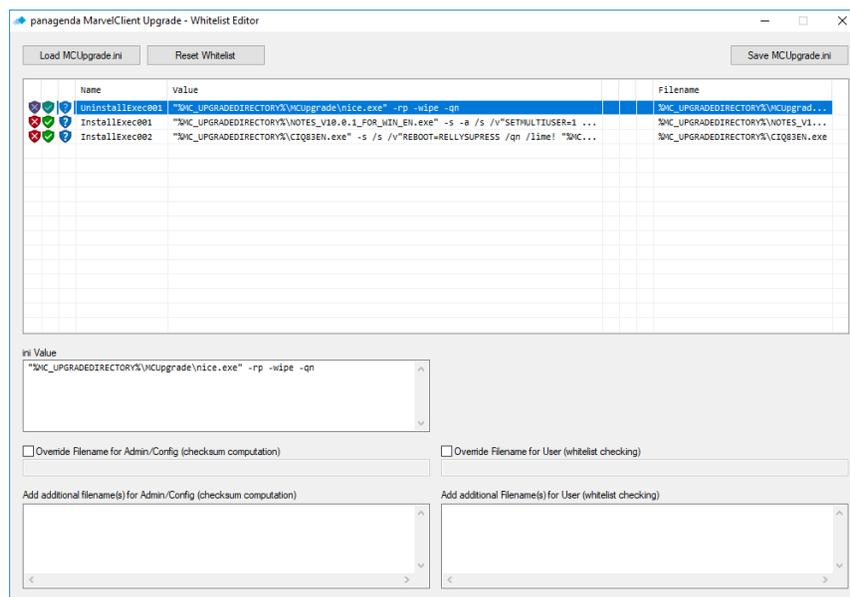


Figure 50: The Whitelisting Editor

Any zip/7z files, installers and executables that the Whitelisting Editor cannot find for Whitelisting are displayed with a red shield in the very first column:

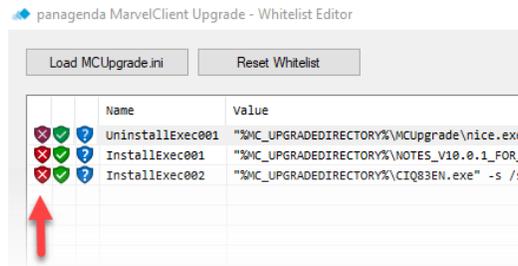


Figure 51: Cannot find source file in Whitelisting Editor

Wrong paths can be corrected for both Whitelisting and later execution on end user workstations using the “Override ...” fields at the bottom of the window:

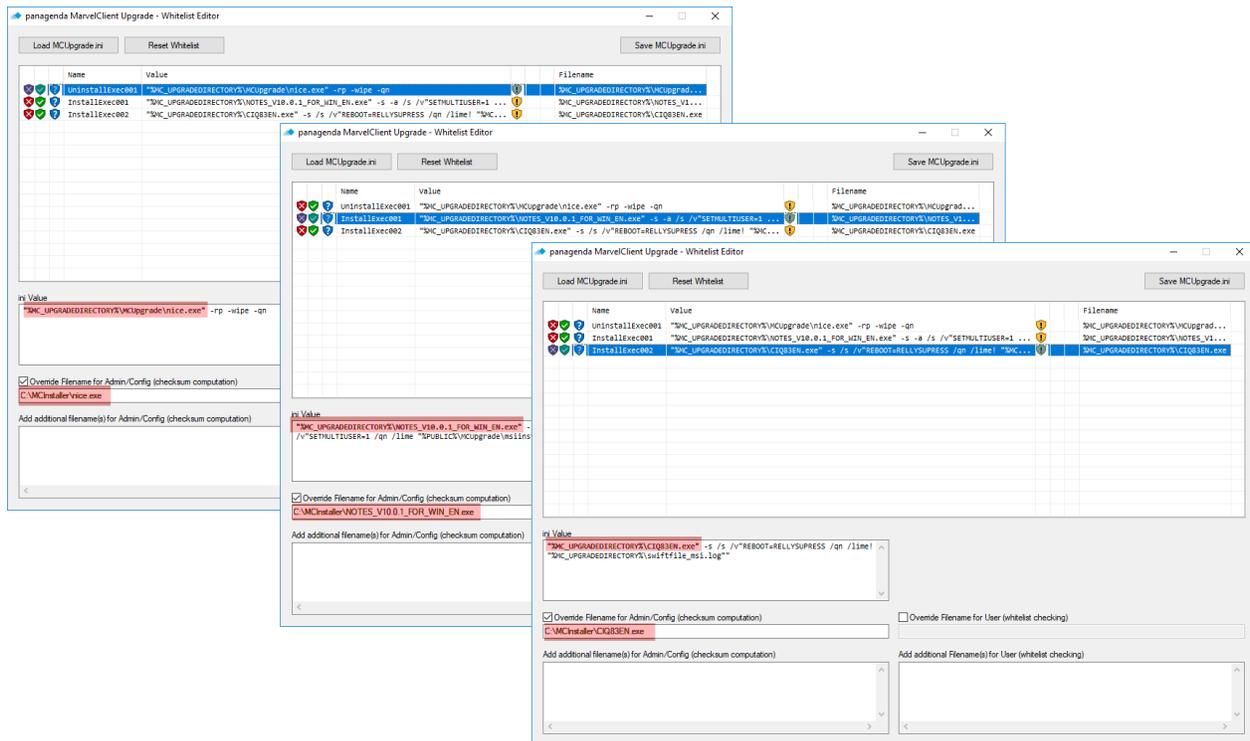


Figure 52: Overriding source files in Whitelisting Editor

After checking all file paths and instructions save your MCUUpgrade.ini file:

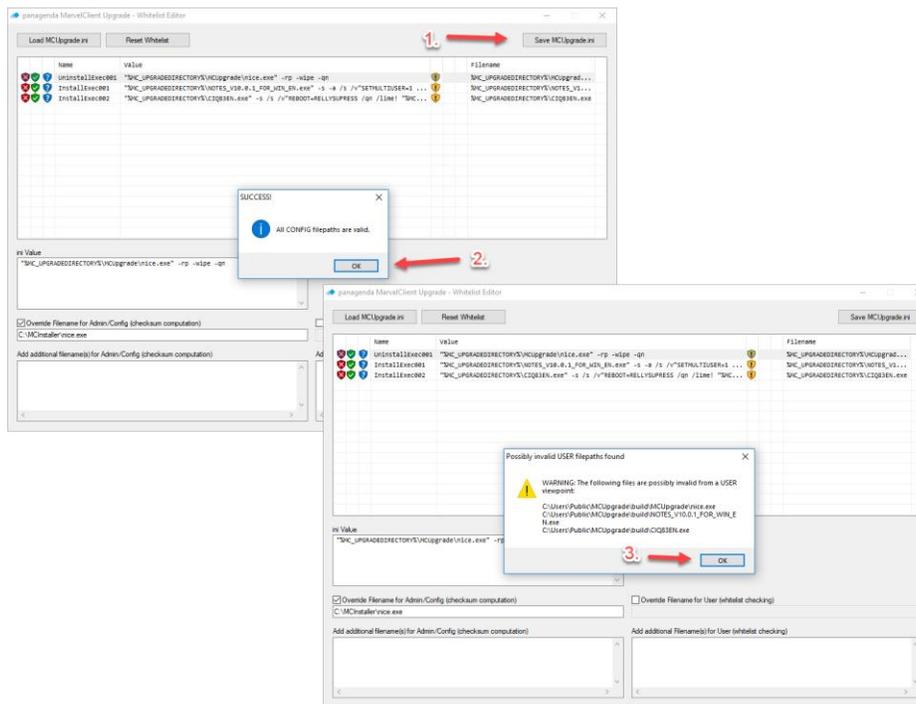


Figure 53: Saving MCUUpgrade.ini Whitelist

Upon saving the whitelist (1), ALL (Admin) CONFIG filepaths must be valid (2) at that exact moment in time. Invalid file paths from a USER perspective (3) can be ignored when preparing/publishing your upgrade.

Saving the MCUUpgrade.ini will hash all zip/7z files, installers, executables and sign a variety of MCUUpgrade.ini entries, to ensure that they cannot be tampered with on end user workstations.

Afterwards you can close the Whitelist Editor.

The preparatory activities have thus been completed. Your package can now be distributed and executed on all workstations on which the panagenda Windows service is installed.

iv. Whitelisting Verification (optional)

If you want to check whether whitelisting was successful you can open the "MCUpgrade.ini" file in the MarvelClient Upgrade Build Directory (%public%\MCUpgrade\build\MCUpgrade.ini) with a text editor. After successful Whitelisting, you will find a "Whitelisting" section (depending on your configuration also a "MCKeyfileEditor" section) at the end of the file:

```

[Whitelisting]
UninstallExec001Check=MC_UPGRADEIRECTORY%\MCUpgrade\nice.exe
UninstallExec001IniChecksum=Vk+gfB35+C8IX5/nbPZq14atzIv5nTsw0dhE5jy8BkeZb1Wo51VSLCgowf5HLxEPcavDPPUkZhF09t2/ZZtM1viSxgFN4+0sQ7eh+FA1tTLgsepUzU2vZobmFMXkPb5zdMMWGAWRGnAqexpInC75Ye6XZK2
UninstallExec001FileChecksum=2489BA95F8546E6049303F74DFCE0FC2538CEB7F02CBF01A0B3819A1F6AC98DE
InstallExec001Check=MC_UPGRADEIRECTORY%\NOTES_V10.0.1_FOR_WIN_EN.exe
InstallExec001IniChecksum=t8m1CDms/D+mF5JAsGjYjuc/L5rYU6Z++5FEhk+JK00TK0fW0TPAwbcdd6GkLSBICGrA3ok05+9/orcH1rizibw1Nb+/fkd4jsaVe99P51wAsv0gbSh800Y1xrBxQCZpde61h2pR0LR/askECn8oWMD08nBxZ3V1
InstallExec001FileChecksum=55E0CFEA5FC1A7A55B4F948214AC99B29E098A29EFFC72144AFB950B59950546
InstallExec002Check=MC_UPGRADEIRECTORY%\CIQ083EN.exe
InstallExec002IniChecksum=u9tn63/tnM6qImxfS91rL15CZIX0G8/5L5xLIwwV6Hpet5Ne1//WvVYTHt7vqpxppWyGA7cE081AgRVItzfaoyFwNr1eN+/sb5qHPqGf11IjTgTn2zyc6YPwm1uRI80/06D0Eh7ZqmMcpdnXmTnSqqY4suOmwR0
InstallExec002FileChecksum=F5942295E8E3C599A3EA7D337911E7E87DE932C358439046439BC3842170DF92
Signature=Y8undyFwW5+ZdGGI2w841QtorIHqysjLXpq56Mgq9H0FR14Uw/KTRK1Hr1cXatg9z9I9IsHP6shA8hwYQ8vY26K0J8qvL81jH1TRcKsnBS1asW433Vhn+c630WJ55PQca8Hh393JernT707wnI2WzY9GRuyAb5Sww2TS+v6M1uFDg8bEy
[MCKeyfileEditor]
UninstallExec001=C:\MCInstaller\nice.exe
InstallExec001=C:\MCInstaller\NOTES_V10.0.1_FOR_WIN_EN.exe
InstallExec002=C:\MCInstaller\CIQ083EN.exe

```

Figure 54: Whitelisting and MCKeyfileEditor sections in MCUpgrade.ini

3.5. SMTP Server Authentication

As you can see from the options on the Tracking tab in your upgrade configuration document, you can track your upgrade via SMTP emails.

If your Domino/mail server requires authentication for sending emails, you can add the corresponding authentication details to your MCUpgrade.mcu file:

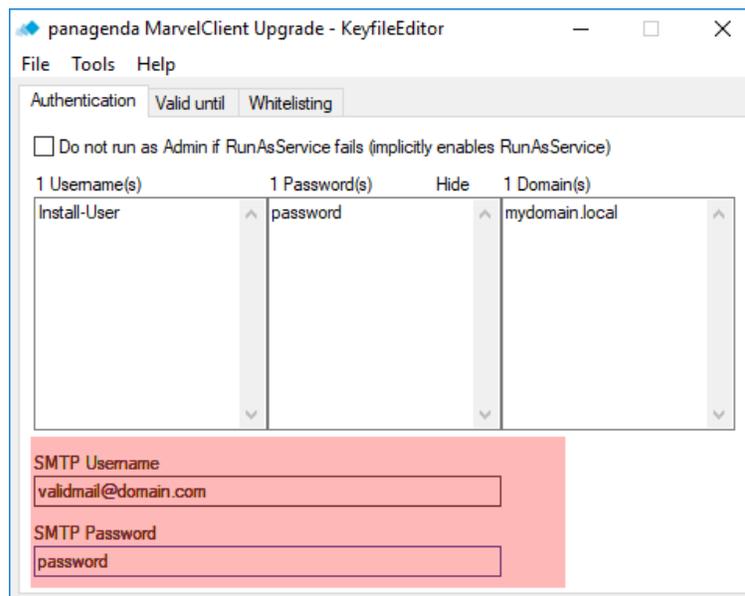


Figure 55: Storing SMTP authentication details in your MCUpgrade.mcu file

Don't forget to save your MCUpgrade.mcu file when you are done.

4. How to setup the panagenda install service

Requirements:

- Windows 7 or 10, 32 or 64 bit
Note: the Notes 10 client is only supported on Windows 7 64Bit and Windows 10 64Bit.
- A software deployment tool or local administration rights

Components:

- Script: Install_MCUUpgrade-Windows-Service.cmd
- DLLs: MCUUpgrade.dll / MCUUpgrade40.dll
- Executable: MCUUpgradeService.exe
- Config: MCUUpgradeService.exe.config
- PublicKey: MCUUpgrade_PublicKey.sig (see 3.4.ii)

All components can be downloaded when selecting “MCUpgrade Service” on the Elevation tab and clicking on “Save Service Install Files”:

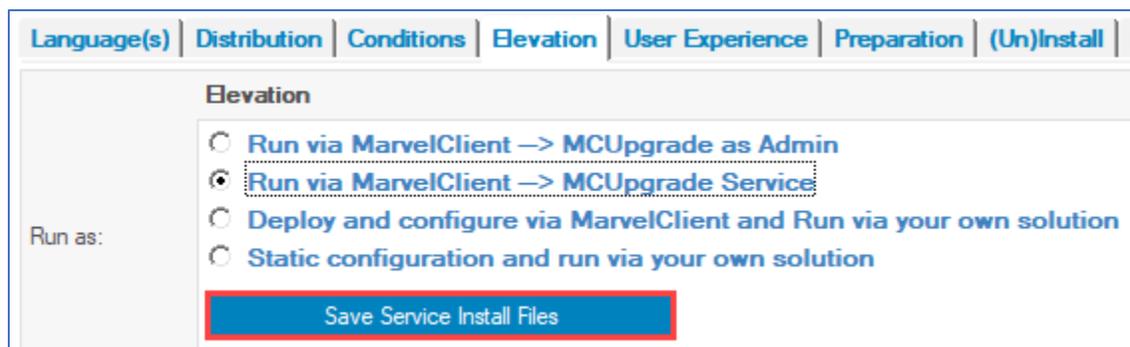


Figure 56: Saving the Service Installer components

Unzip the resulting file to a folder of your choice:

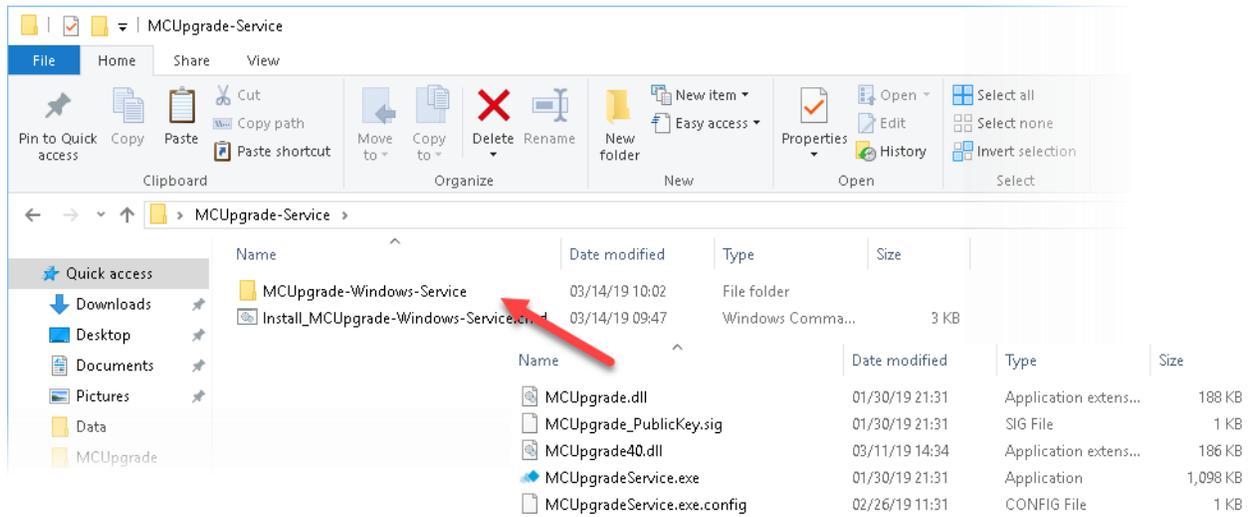


Figure 57: Unzipping the Service installer zip file

Important:

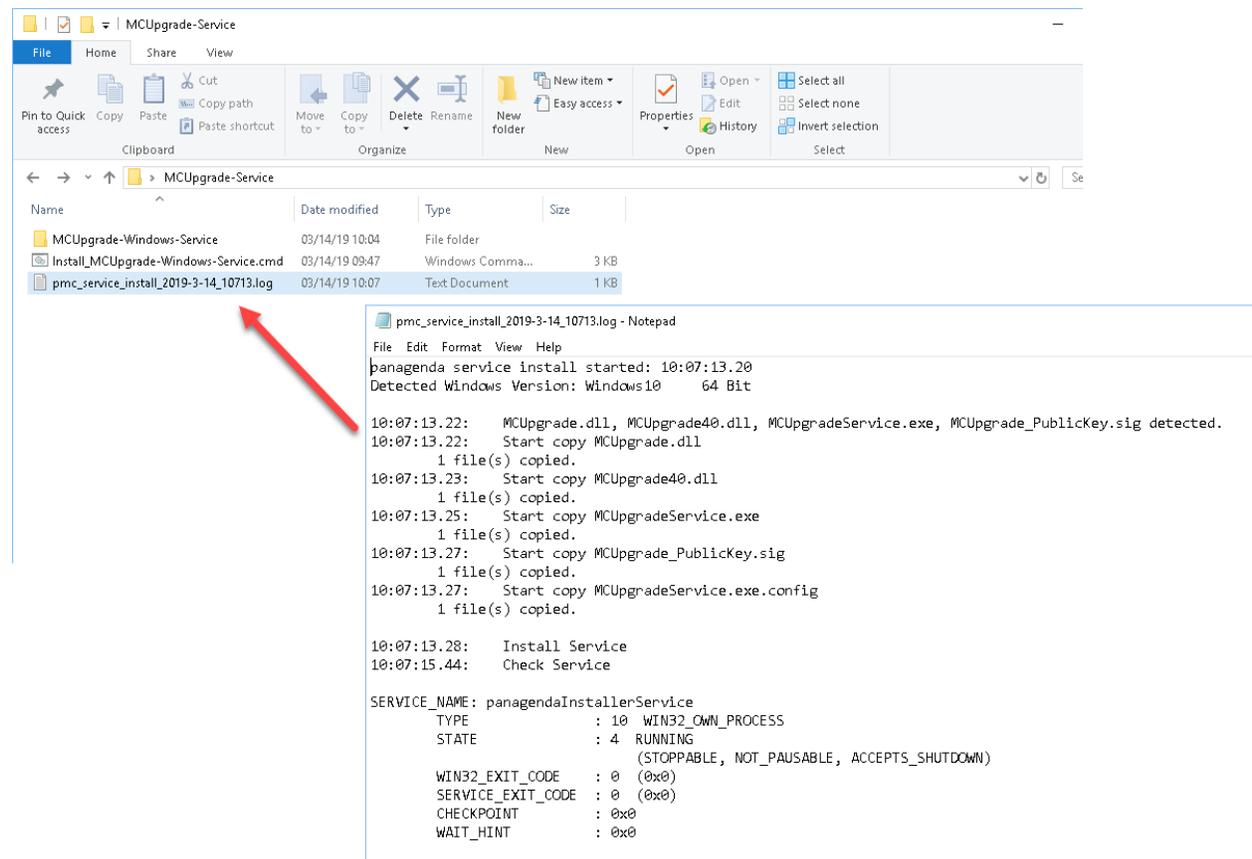
Before you continue, please make sure that you add your MCUpgrade_PublicKey.sig file to the MCUpgrade-Windows-Service subfolder.

4.1. Automatic Installation

Simply execute the supplied install script “Install_MCUpgrade-Windows-Service.cmd” with an admin account or your software deployment solution, which will install and start the service.

Tip:

During installation, a logfile is created within the execution folder:



The screenshot shows a Windows File Explorer window for the folder 'MCUpgrade-Service'. The file list includes:

Name	Date modified	Type	Size
MCUpgrade-Windows-Service	03/14/19 10:04	File folder	
Install_MCUpgrade-Windows-Service.cmd	03/14/19 09:47	Windows Comma...	3 KB
pmc_service_install_2019-3-14_10713.log	03/14/19 10:07	Text Document	1 KB

A red arrow points to the log file. The Notepad window shows the following log content:

```

panagenda service install started: 10:07:13.20
Detected Windows Version: Windows10 64 Bit

10:07:13.22:   MCUpgrade.dll, MCUpgrade40.dll, MCUpgradeService.exe, MCUpgrade_PublicKey.sig detected.
10:07:13.22:   Start copy MCUpgrade.dll
                1 file(s) copied.
10:07:13.23:   Start copy MCUpgrade40.dll
                1 file(s) copied.
10:07:13.25:   Start copy MCUpgradeService.exe
                1 file(s) copied.
10:07:13.27:   Start copy MCUpgrade_PublicKey.sig
                1 file(s) copied.
10:07:13.27:   Start copy MCUpgradeService.exe.config
                1 file(s) copied.

10:07:13.28:   Install Service
10:07:15.44:   Check Service

SERVICE_NAME: panagendaInstallerService
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 4   RUNNING
                        (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE       : 0   (0x0)
        SERVICE_EXIT_CODE   : 0   (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0
  
```

4.2. Manual Installation

Step 1: Create the following folder on the destination computer
“C:\Program Files (x86)\panagenda\MCUpgrade”

Step 2: Copy the following files into the folder from step 1:

1. MCUpgrade.dll
2. MCUpgrade40.dll
3. MCUpgrade_PublicKey.sig
4. MCUpgradeService.exe
5. MCUpgradeService.exe.config

Step 3: Open a command console (cmd) and change to the folder from step 1.

Important: You must have admin rights when opening the command console.

Then execute the following line:

```
MCUpgradeService.exe install start --localsystem --sudo --autostart
```

The Service will be installed and started.

On every computer, where the "panagenda Installer Service" is in place, MCUpgrade will work without having to worry about user names and passwords or UAC settings.

Name	Description	Status	Startup Type	Log On As
 panagenda Installer Service	panagenda ...	Running	Automatic	Local System...

Figure 58: The panagenda Installer Service

5. Security Considerations

5.1. Local built-in Administrator, local or Domain Admin

Your MCUUpgrade.mcu keyfile, which contains the admin username(s) and password(s), is deployed on all clients to switch to elevated mode and is encrypted using the Advanced Encryption Standard 256 Bit (AES 256). If so selected in the upgrade wizard (Data Migration tab, Cleanup section, default=yes), the file is deleted at the end of a successful upgrade.

During elevation the usernames and passwords are used to execute RunAs commands.

5.2. Using the MarvelClient Upgrade Service

The service only elevates executables that are signed with an official Microsoft EV (Extended Validation) certificate from panagenda. Any other executables will be rejected. Windows Installers are not affected by this, as they are executed after elevation and checked against the whitelist.

Whitelisting is secured with your own RSA key pair, using RSA 4096 bits.

Whitelisting ensures that MCUUpgrade only executes install and uninstall instructions “signed” by you and for which the Secure Hash Algorithm SHA256 has not been tampered with.

5.3. Using your own corporate software deployment with MCUUpgrade

Execution safety is entirely subject to the make and design of your own software deployment solution.

Running MarvelClient Upgrade via your own software deployment solutions come with the following requirement and important notes:

- MCUUpgrade cannot be run before Windows login and requires a logged in user. That is because of various dependencies that can only be resolved via a user session, such as environment variables and folder locations, for example, which are very different when running in system context exclusively.
- The Notes client must not be running. You can run “MCUUpgrade.exe /shutdown” (requires user interaction) or “MCUUpgrade.exe /shutdownsilent” (will end all Notes client components by automatically answering message boxes) in user context to attempt proper closing of an up and running client

6. Table of Figures

Figure 1: Online Update settings for MarvelClient Essentials customers	8
Figure 2: Create a new "Upgrade Configuration" action	9
Figure 3: General part of "Upgrade Configuration" action	10
Figure 4: Upgrade Package Distribution	11
Figure 5: Import Full Client Installer 1/4.....	11
Figure 6: Import Full Client Installer 2/4.....	12
Figure 7: Import Full Client Installer 3/4.....	12
Figure 8: Import Full Client Installer 4/4.....	13
Figure 9: Import additional Installers 1/4	13
Figure 10: Import additional Installers 2/4.....	13
Figure 11: Import additional Installers 3/4.....	14
Figure 12: Import additional Installers 4/4.....	14
Figure 13: Conditions tab	15
Figure 14: Elevation tab using local built-in administrator account.....	16
Figure 15: User Experience tab.....	17
Figure 16: Uninstall.....	18
Figure 17: Full Client Installation command line.....	18
Figure 18: "Fixpack / Hotfix / Language Pack(s) / other Installation" command lines 1/2.....	19
Figure 19: "Fixpack / Hotfix / Language Pack(s) / other Installation" command lines 2/2.....	19
Figure 20: Tracking tab (SMTP configuration).....	20
Figure 21: Prepare or Publish Upgrade 1/3.....	22
Figure 22: Prepare or Publish Upgrade 2/3	22
Figure 23: Prepare or Publish Upgrade 3/3.....	22
Figure 24: MarvelClient Upgrade Actions.....	23
Figure 25: The MCKeyfileEditor.....	25
Figure 26: The Whitelisting Editor.....	26
Figure 27: The Local Built-In Admin Account	27
Figure 28: Launching MCKeyfileEditor.....	27
Figure 29: KeyfileEditor Admin Credential Input	28
Figure 30: Saving your MCUUpgrade.mcu file 1/2	28
Figure 31: Saving your MCUUpgrade.mcu file 2/2	29
Figure 32: Attaching your MCUUpgrade.mcu file to your upgrade configuration	29
Figure 33: Providing Local or Domain user credentials	30
Figure 34: Using multiple usernames and passwords in MCKeyfileEditor.....	31
Figure 35: Using multiple passwords with the same user in MCKeyfileEditor.....	31
Figure 36: UAC Admin Approval Mode	33
Figure 37: Admin Approval Prompt for consent	33
Figure 38: UAC: Prompt for consent	33
Figure 39: Local Security Policy	33
Figure 40: ConsentPromptBehavior Registry setting	34
Figure 41: Launching the MCKeyfileEditor to create your keypair.....	35

Figure 42: Create keypair menu option in KeyfileEditor.....	36
Figure 43: Your keypair filenames	36
Figure 44: Message at end of keypair generation.....	36
Figure 45: Exclusively using the Windows service.....	37
Figure 46: Missing Whitelist warning.....	38
Figure 47: Securing your MCUUpgrade.mcu keyfile with a password	38
Figure 48: Saving your MCUUpgrade.mcu keyfile	38
Figure 49: MCUUpgrade.mcu and MCUUpgrade_PublicKey.sig attachments.....	39
Figure 50: The Whitelisting Editor.....	39
Figure 51: Cannot find source file in Whitelisting Editor	40
Figure 52: Overriding source files in Whitelisting Editor	40
Figure 53: Saving MCUUpgrade.ini Whitelist	41
Figure 54: Whitelisting and MCKeyfileEditor sections in MCUUpgrade.ini	42
Figure 55: Storing SMTP authentication details in your MCUUpgrade.mcu file	42
Figure 56: Saving the Service Installer components	43
Figure 57: Unzipping the Service installer zip file	44
Figure 58: The panagenda Installer Service	46

7. 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 Quick Start Guide 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.