

MarvelClient Upgrade





UPGRADE MANUAL

MC Upgrade 7.x

mc.dll >= 6.x

Contact

panagenda Austria

(Headquarters)

panagenda GmbH
Schreyvogelgasse 3/10
AT 1010 Vienna (Austria)
Phone: +43 1 89 012 89
Fax: +43 1 89 012 89 – 15

panagenda Germany

panagenda GmbH
Lahnstrasse 17
DE 64646 Heppenheim
(Germany)
Phone: +49 6252 67939 – 00
Fax: +49 6252 67939 – 16

panagenda USA

panagenda Inc.
60 State Street
Suite 700
Boston, MA 02109 (USA)
Phone: +1 (617) 855 5961
Fax: +1 (617) 488 2292

E-Mail Sales: sales@panagenda.com
E-Mail Support: support@panagenda.com
Web: www.panagenda.com

Table of Contents

1 Welcome to panagenda MarvelClient	5
2 Features	6
3 A true quick start into MarvelClient Upgrade	7
4 Prerequisites	8
4.1 MarvelClient Upgrade checklist/questionnaire	8
5 Upgrade components	10
5.1 Required downloads	10
5.2 Additional, free software	10
6 Online Update	10
7 Example client upgrade	11
8 Configuration instructions	13
8.1 An overview of the necessary steps	13
8.2 Software packaging	17
8.3 Instructions – step by step	18
8.3.1 Zip installation package	18
8.3.2 MCKeyfileEditor.exe	19
8.3.3 MCUpgrade.exe and MCUpgrade.ini	20
8.3.4 Visualizing Progress	21
8.3.5 Parsing install or uninstall logs for errors	21
8.3.6 Status Report Emails and parameters in MCUpgrade.ini	23
8.3.7 Unzipping of compressed archives before uninstalling and installing	23
8.3.8 Uninstalling old IBM Notes client releases	24
8.3.9 Installing a new IBM Notes client release	25
8.3.10 Re-run of unsuccessful upgrades/updates	25
8.3.11 Allowing users to defer an upgrade/update	26
8.3.12 Graceful Shutdown	28
8.3.13 Prepare upgrade components for distribution	29
8.3.14 Smart File Downloader	30
8.3.15 Distribution of upgrade files	33
8.3.16 Language Packs and the notes.ini Region setting	34
8.3.17 Generic customization of MCUpgrade*.ini	35
8.3.18 Launching MarvelClient Upgrade	37

8.4 Useful MarvelClient Features for MC Upgrade	38
8.4.1 Querying IBM Notes client release (text-based)	39
8.4.2 Querying IBM Notes client release (numeric)	40
8.4.3 Checking for sufficient disk space and memory	40
8.4.4 Cleanup of notes.ini after an upgrade	41
8.4.5 Deleting installation packages after an upgrade	41
8.5 Performing an update only	41
8.6 Description of MCUUpgrade.ini entries	43
8.7 Series of events within MCUUpgrade(Main).exe	45
DISCLAIMER	47
8.8 Limitation of liability for external links	47

1 Welcome to panagenda MarvelClient



This guide contains comprehensive documentation to get the most out of MarvelClient Upgrade. If you have any comments or suggestions, please contact us at support@panagenda.com.

panagenda MarvelClient gives you control over all your IBM Notes client configurations ensuring smooth operations all year round. From the initial client setup, unique roaming and continuous implementation of desired settings, to keeping clients up-to-date smartly. Take a look at our other MarvelClient Modules on our webpage www.panagenda.com/products/marvelclient/. The Marvel Client Help database provides documentation for all MarvelClient Modules: www.panagenda.com/webhelp/marvelclient

About MarvelClient Upgrade

Regardless of how your user's clients are configured today, with MarvelClient Upgrade you can upgrade and standardize them reliably and with very little effort!

Primarily, IBM Notes client upgrades are costly and complicated for the following three reasons:

- Select end user configurations differ from your corporate standard. Whilst an upgrade project often works just fine for 90-95% of all users, it fails for the other 5-10%. And it takes as much time to run after those 5-10%, as it takes to upgrade the other 90-95%. Some users have basic clients, others have standard, some have network based data directories, while again others have local data drives with any number of local replicas. And while the majority of your Notes installs reside in a standard directory, some of your users have program and data directories in any place imaginable.
- Your existing corporate software deployment does not reach or work for all of your users. MarvelClient Upgrade can be perfectly combined with corporate software deployment: Smart, network bandwidth preserving deployment of software packages and intelligent, as well as MarvelClient's generic and auto-adapting installation process can be mixed and matched with your existing tools.

- Other software deployment tools and the IBM installers (whether used as is or tweaked with e.g. Installshield Tuner) are “not IBM Notes aware”. For example, they cannot cope with changing multiple existing data directories for multi-user installations on the same computer (think changing the data directory from *\\Lotus\\Notes* to *\\IBM\\Notes*), and they are unable to migrate data directories from network drives to local disks. MarvelClient Upgrade makes all the difference, allowing you to address virtually any upgrade and update scenario without the limitations of classic software deployment tooling.

Move beyond traditional upgrade tools where an upgrade package is pushed out whilst the upgrade team waits for feedback and hopes for the best. With MarvelClient Upgrade you can see the configuration and status of your client upgrades and downtime can be avoided with flexible actions that adapt to each client and user independent of when changes occur.

2 Features

The most important features of MarvelClient Upgrade are:

- Intelligently distribute installation packages to clients with smart network load reduction
- Easily upgrade, update, downgrade and crossgrade your IBM Notes clients
- Flexibly change any existing installation type to any other (e.g. single-user ⇔ multi-user), including support for migrating multiple existing data directories on the same computer, as well as migrating data directories from network drives to local disks
- Conveniently install and uninstall Fix Packs, Feature Packs Language Packs, IBM Sametime embedded, IBM Connections Plugins, and more
- Install and uninstall any other software
- Stay informed with Status Report Emails
- Prevent error-prone scripting with MarvelClient’s unique, flexible and fault-tolerant upgrade engine
- Enjoy smart data-loss prevention: open clients with unsaved documents are detected automatically, allowing your end users to save them (see “Graceful Shutdown” on page 28)

3 A true quick start into MarvelClient Upgrade

Whilst this upgrade guide has 47 pages, the overall principle of MCUUpgrade can be summarized as follows:

First, download the MarvelClient Upgrade components and perform a onetime creation of an encrypted keyfile with admin and SMTP credentials.

Thereafter, for any upgrade/update/downgrade/crossgrade, follow these five easy steps:

Download IBM installer(s)

Improve installer compression

Configure/Finetune your upgrade/update

Deploy compressed installer(s)

Run upgrade/update/downgrade/crossgrade

Especially upgrades can come in many different shapes and forms. MarvelClient Upgrade empowers you to cope with ANY mix of installations, operating systems, languages and setups in your existing environment with a single upgrade path configuration.

Whilst your very first upgrade/update configuration can take a day or two, subsequent upgrade and updates are usually done within an hour or two, as they only require exchanging installer files and finetuning of your existing configuration.

To get the most out of MarvelClient Upgrade, the following chapters describe all available options and features in detail – many of which you won't need to follow above easy steps, due to the built-in defaults and provided examples. Yet, the following pages are designed to help you to understand the full power of MarvelClient Upgrade.

If you want to get started fast without reading all the following chapters, you may wish to explore the MCUUpgradeExample* files, that are included in your MC Upgrade download.

In conjunction with appropriate training we are also happy to provide quickstart videos to you that guide you through how our consultants go about creating update and upgrade configurations – just contact us for more information.

4 Prerequisites

To get the most out of MarvelClient Upgrade, the following are helpful, but all in all non-mandatory prerequisites:

- Installing MarvelClient on your IBM Notes clients
This is easily done through installation via Email or a PostOpen script in the mail template, or through your existing software deployment
- Basic experience with MarvelClient to understand logical relations and sequences of Actions, Conditions and Access Definitions
- Basic experience with MarvelClient String and Numeric Conditions
- Basic experience with MarvelClient Access Definitions
- Basic experience with MarvelClient Actions of the following types:
 - Run Program
 - File Deployment
 - Smart File Downloader
 - *.ini & Variables
- A compression/zip tool (see: "Upgrade components" on page 10)
- In the end, all you need to make your next upgrade a success, are the panagenda MarvelClient Upgrade components "Upgrade components" on page 10) together with a license file

4.1 MarvelClient Upgrade checklist/questionnaire

1. Is MarvelClient currently deployed on all users' IBM Notes clients?
If so – or if you plan to invoke your MarvelClient Upgrade with your own software deployment tooling – proceed to 2., if not, deploy to all users first
2. Are you deploying a Hotfix, Fix Pack, Feature Pack, Language Packs, Social components, Sametime embedded or similar?

Note down the logical order in which such components need to be installed:

3. Where are the users to be upgraded/updated physically located?
This is only relevant if you wish to copy the upgrade package onto file shares or a web server in different geographical areas. If it makes sense to copy the upgrade package to file shares or web servers, list them here:

4. Prepare that you – or a colleague – can supply the local and/or domain Admin ID(s) and password(s) for the encrypted keyfile of MarvelClient Upgrade. This is later required for the necessary administrative access during upgrades and updates.
For the status report emails, SMTP access must be provided. Please have your access information ready for this.



TIP | ***Do NOT note security related information down here as the account and password should never be available anywhere as plain text!***

5. Pick a target directory on the client workstations that all end-users have read/write access to, such as <env:public>\panagenda\MCUUpgrade\ and note it here for later use

5 Upgrade components

MarvelClient Upgrade primarily consists of *.exe files for the processing of the actual install and uninstall logic. These *.exe files are either invoked via MarvelClient Actions from within IBM Notes, or through corporate software deployment. The MarvelClient Upgrade license check is performed at runtime by searching for license_upgrade.xml in the UpgradeDirectory.

5.1 Required downloads

- ▶ Your personal *MCUpgrade_maj_min_step_rev.zip* contains all necessary upgrade components. For example: *MCUpgrade_7_3_0_5191.zip*

Your personal download link can be found on the bottom right of the following page:

<https://update.panagenda.com/pub/panaweb.nsf/GetLicenseInfo?openagent&key=YOURLICENSEKEY>



Please contact support@panagenda.com if your individual, license-based download link is not yet available on above page.



*Note: Some browsers/proxies block the download because the zip archive contains *.exe files*

5.2 Additional, free software

- ▶ **7-ZIP** – Very efficient compression and decompression software for reducing the size of you upgrade packages

<http://www.7-zip.org/>

6 Online Update

The Action documents mentioned in this documentation are available as Examples in the MarvelClient Configuration database. An Online Update should be conducted in order to ensure that all documents are compatible with the newest version of MC Upgrade.

7 Example client upgrade

This example outlines the individual steps in an upgrade in which an IBM Lotus Notes 8.5.3 single-user installation is updated to an IBM Notes 9.0.1 multi-user installation. Note that in case you only wish to install an update (such as a Fix Pack, Feature Pack, update Sametime embedded or similar) you should still read through the complete upgrade instructions to fully understand MarvelClient Upgrade. After doing so, you can then limit your configuration according to chapter “Performing an update only” on page 41.

1. Unless you plan to invoke MarvelClient Upgrade through your software deployment tool, make sure that MarvelClient (MC) is installed and running on all clients that are to be upgraded.
2. Using MC Smart File Downloader, all clients download an installation package for either file or web servers, containing IBM Notes 9.0.1, language packs, Feature Pack 10 and Sametime 9.0.1 embedded Hotfix – all without bloating the network, but by load-balancing downloads over time and across your clients. To make the download as small as possible (every MB counts when multiplied with the number of targeted users), all installation files and folders are packed into a single 7-ZIP archive.
3. The upgrade process is either initialized via software deployment or from within IBM Notes in the background and/or during client shutdown. When integrated in IBM Notes, end users can optionally be prompted for whether s/he wishes to start the upgrade and postpone according to flexible configuration options. Also, integration with IBM Notes checks for open and unsaved documents and does not invoke an upgrade before all documents have been saved or closed by your end users.
4. As soon as the user confirms the upgrade or the deferral period has ended, and all unsaved documents are saved, upgrade begins to run in end user context first.
5. In end user context, the old Notes data directory is copied to the new data directory, if it was previously stored on a network drive, and the new data directory is in a different location. This is because usually Admin IDs or the System Service do not have access to end user network drives.
6. Next, MC Upgrade switches to the provided Admin context using credentials as provided in encrypted format in the MCUpgrade.mcu keyfile; this process includes UAC elevation.
7. Upon successful switch to one of the provided Admin credentials, any shortcuts to IBM Notes (including Admin and Designer client and other relevant shortcuts) are removed to prevent users from accidentally starting Notes during an upgrade. Optionally, the direct launching of e.g., notes.exe, nlnotes.exe and similar executables can be prevented during an upgrade, too.

8. Subsequently, any zip/7z files provided for the upgrade/update will be unzipped.
9. Then, the previously existing client is uninstalled. Any personalized client data (=the IBM Notes data directory) remains unaffected by the uninstall process.
10. After uninstalling the existing client, any old data directories are migrated to the new data directory structure, if different. In case of e.g. multiple c:\...\lotus\notes\data directories, for example, this can be automatically migrated by MC Upgrade to c:\...\IBM\Notes\Data. Similarly, previous single user data directories are migrated to multi-user data directories and vice versa, according to your particular upgrade case.



NOTE that with MC Upgrade, notes.ini files are automatically adjusted during the data directory migration process, too. If desired, mc.dll (MarvelClient extension) can also be automatically integrated into the further-on newly installed client.

11. Any files and directories remaining from the uninstall process, that are no longer required, will then be deleted by MarvelClient Upgrade (again, this does not affect personal data).
12. Next, the installation of the new IBM Notes client is carried out in the following sequence:
 - a) Installation of IBM Notes 9.0.1 multi-user
 - b) Installation of language packs
 - c) Installation of Feature Pack 10 Interim Fix 1
13. The installation is complete and the new multi-user IBM Notes client can be started automatically by MarvelClient Upgrade or manually

8 Configuration instructions

Administrators who intend to configure MarvelClient for the IBM Notes client upgrade should first install the software tools referred to above on their workstation. These instructions assume that all requirements are met and recommendations followed. Accordingly, 7-ZIP or any similar compression tool for creating zip or 7z files must be installed.

8.1 An overview of the necessary steps

The following table lists the steps necessary for performing a client update/upgrade:

- ▶ An **Upgrade** refers to moving from e. g. IBM Lotus Notes 8.5.3 to IBM Notes 9.0.1 or similar (=Uninstall/Install),
- ▶ An **Update** refers to installing a Feature Pack, Fix Pack, Hotfix, Sametime embedded, IBM Connections plugins and similar.



Note that it is advisable to at first limit ANY/ALL MC upgrade Actions to a (group of) test user(s) until all results are satisfactory.

Configuration Steps		
	Upgrade	Update
Download installers from IBM	yes	yes
Since MarvelClient Upgrade 5.4.0, customizing the IBM installer to include mc.dll and adjust notes.ini is no longer necessary. Instead, simply include mc.dll into the MCUpgrade.zip file (see below) as follows: Add mc.dll to EITHER a subfolder CopyToProgramDirectoryNew or CopyToMultiUserSharedDataNew\Common, depending on whether it shall in the end reside in each users' program or data directory. Any files placed in these directories will be copied into the NotesProgram or MultiUserSharedData\Common directory after installing the new client. Similarly, add a custom notes.ini file into EITHER the CopyToProgramDirectoryNew (for single user clients), or CopyToMultiUserSharedDataNew (for multiuser clients). Any *.ini, *.properties and *.txt files will be MERGED into the respective target files after installing the new client. For *.ini files, please do not forget to use the needed section(s) - for notes.ini it is [Notes].	yes	no

Table 1: Necessary Steps – Overview

Configuration Steps		
	Upgrade	Update
<p>Zip up any setup files into one or multiple 7z/zip archives with compression level set to "Ultra/Highest"</p> <p>For example: one *.7z file with subfolders NOTES901, FP10IF1 and language pack, and another *.7z file with just FP10IF1 for clients that are already 9.0.1 but don't have the Feature Pack yet.</p> <p>It is not advisable to create one *.7z with just 9.0.1 without Fix Pack and another with just the Fix Pack in order to target some clients with two packages and other clients with one. It is easier to have one single complete package as per upgrade/update path.</p>	yes	yes
<p>Create an <i>MCUpgrade.ini</i> as per upgrade path, for example:</p> <p>"MCUpgrade_NOTES901.ini"</p> <p>"MCUpgrade_NOTES901FP10IF1.ini"</p> <p>► If you only have one upgrade path, you can simply change <i>MCUpgrade.ini</i> as needed and continue to use that</p> <p>Please refer to "Instructions – step by step" on page 18 to get more information about configuring <i>MCUpgrade.ini</i> according to your individual requirements.</p>	yes	yes
<p>Test the upgrade manually by</p> <ol style="list-style-type: none"> providing (a) test machine(s) on which you have a representative installation that should be upgraded/updated making sure you can revert back to the state of the machine from before attempting the upgrade making sure the upgrade runs successfully and WITHOUT any end-user interaction <p>Depending on the Upgrade/Update and source packages provided by IBM, this can be accomplished by using either</p> <ul style="list-style-type: none"> • <i>Setup.exe</i> together with whatever parameters make sense, or • <i>Msiexec / msifile.msi</i> with whatever parameters, or • <i>Msiexec / msifile.msi transform=mstfile.mst</i> with whatever additional parameters <p>Simply run the upgrade from a command line (<i>cmd.exe</i>). Where necessary launch <i>cmd.exe</i> with administrator rights</p> <p>If the upgrade/update works as expected and without any end-user interaction continue with the next step</p>	yes	yes

Table 1: Necessary Steps – Overview



Configuration Steps		
	Upgrade	Update
<p>Using MCKeyfileEditor.exe, create your own keyfile (e.g., MCUUpgrade.mcu) which stores the access information for your administrator account(s), as well as the SMTP account.</p> <div>  <p>Note that your keyfile is password protected with a password of your choice for both reading and editing, and multi-level encrypted with AES-256 bit.</p> <p>Also note that in theory you will only need to do this one single time as the keyfile can be reused for as many upgrades and updates as needed in the future, provided that the accounts/passwords/domain do not change</p> </div>	yes	yes
<p>Zip up the following files into one single zip file called "MCUUpgrade.zip": all "MCUUpgrade.ini" files, your MCUUpgrade.mcu file, license_upgrade.xml and MCUUpgrade.exe (you can naturally choose for another zip file name – this filename, however, is referred to later in this documentation) – don't forget to include the CopyTo... subfolders, if applicable.</p>	yes	yes
<p>Create a File Deployment Action in the MarvelClient Configuration database to distribute MCUUpgrade.zip; using appropriate target users, groups and/or Conditions you may want to limit this to only those clients that are to execute (one of) your upgrade path(s).</p> <p>The target directory for the FileDeployment Action is your upgrade directory, e.g., <env:PUBLIC>\panagenda\MCUUpgrade</p> <p>Close and restart your test client – this should lead to the deployment and automatic unzipping of the above MCUUpgrade.zip into the upgrade directory.</p>	yes	yes
<p>Create (a) Smart File Download Action(s) to deploy the install package(s) to the right clients</p>	yes, if not deployed with own tooling	yes, if not deployed with own tooling
<p>Create an Action to configure the Smart File Downloader</p>	optional, for testing	optional, for testing
<p>Test the Smart File Downloader ⇒ restart a test client and make sure it downloads the correct install package according to your configuration</p> <div>  <p>Note that during download the target filename is "the_filename_you_configured.tmp" – as soon as the download is finished, the ".tmp" extension is removed</p> <p>Also note that the default Smart File Downloader configuration waits 5 minutes after client startup plus a random number of minutes (which could be hours) if you configured the Smart File Download Action to only download a couple of chunks in a larger time window. For testing you may want to simply configure it to download 100,000 chunks.</p> </div>	optional, for testing	optional, for testing

Table 1: Necessary Steps – Overview


Configuration Steps		
	Upgrade	Update
<p>If you also want to test deferrals (=allowing end-users to choose for a good time to perform an upgrade or update), make sure that your test client has fully downloaded the install package that matches the corresponding <i>MCUpgrade.ini</i> deferral parameters (if applicable).</p> <p>If you don't want to wait for Smart File Downloading (aka you have successfully tested Smart File Downloading earlier), you can naturally also copy the necessary install package onto the test machine into the upgrade directory.</p> <p>Close your test client – if everything is properly configured, MCUpgrade.exe component should detect that the file download has completed and prompt the end-user whether s/he wants to upgrade/update. Fine tune deferral as needed.</p> <div>  <p>Once an upgrade/update is postponed for the first time, a memory file <i>MCUpgrade_deferral.ini</i> is created locally in the upgrade directory. The content of this file is quite easy to interpret:</p> <pre>[MCUpgrade] DeferralPromptFirst=20131010 DeferralPromptCount=2 DeferralPromptLast=20131011</pre> <p>Modifying this file enables you to reset the memory of the deferral logic.</p> </div>	optional, only if end-users shall be able to defer an upgrade	optional, only if end-users shall be able to defer an upgrade
Now create an MC Run Program Action to execute the actual upgrade.	yes	yes
After having verified that all of the above steps work as expected, change any testing configuration back to production. Remove the download file and the <i>notes.ini</i> variable triggering the upgrade/update from your test client, restart it and make sure it now downloads like you would want to see it in production. Since this may take hours, depending on your configuration, you can revisit this step from time to time.	yes	yes
In case you have more than one upgrade path, make sure that all possible combinations of old/new version are properly configured for and tested	yes	yes
Last but not least you may want to configure for cleaning up the update directory as soon as the newly upgraded/updated client is started so as to free up disk space, that is otherwise occupied by the (unzipped) install packages.	optional, if not advisable	optional, if not advisable

Table 1: Necessary Steps – Overview

8.2 Software packaging

This manual does not deal with setting up and packaging IBM Notes client installation packages and their configuration with *InstallShield Tuner*. There are many tutorials to be found online on how to use *InstallShield Tuner* to create so called transform-files to customize (not just IBM Notes) install packages. Note that *InstallShield Tuner* is free for IBM Notes customers, but only for customizing IBM Notes install packages. If you want to create install packages for Office products or similar, you need to have/obtain a separate license.

For mere learning purposes, the video "MC Integration" referred to above (see: "Additional, free software" on page 10) shows an example of how to add the *mc.dll* into an IBM Notes client install package, as well as how to add entries to the root *notes.ini* file. Any customization of the install package is entirely optional, as in most cases the installer can usually be sufficiently controlled via command line parameters.

For ease of use, the following *notes.ini* parameters should simply be added to the CopyTo...\notes.ini as referred to in row two of the table in section 7.1:

[Notes]

EXTMGR_ADDINS=mc.dll

MC_AllowSyncDuringSetup=2

MC_DB=%notes_homeserver%!!\panagenda\panagenda.nsf



Note: Adjust *panagenda\panagenda.nsf* in the *MC_DB* line as needed.

Further recommended *notes.ini* parameters:

MC_msg_suppress_0d4e=1

(suppresses the "Notes configuration has been completed" message)



Note: This is only needed if you use Lotus Notes 6/7 or >8 and if you use it in Basic mode or ICAA. On a IBM Notes Standard client it is not needed.

MC_DB_NotResolvable=YourDominoserver!!\panagenda\panagenda.nsf



Note: replace "YourDominoserver" with a fixed server name on which the MC Configuration database can be found in case the users mail server cannot be resolved upon first client startup (for example because the user has no location at all or the MailServer is not yet set).

It is advisable to also add:

Ports=TCPIP

and

TCPIP=... according to what your default TCPIP Port settings are to be set to, so that the root *notes.ini* already has a properly configured port to communicate via during the first client start.



Note: make sure that you get the **TCPIP=** configuration right (make use of port compression, for instance);

Example:

TCPIP=TCP,0,15,0,,12320.



If you are already using **MarvelClient Roaming**, don't forget to also add the following parameter:

MC_NetworkDirectory=<drive:>\<path of backup set>

Example: **MC_NetworkDirectory=H:\Notes_Backup**

Note that this can also point to a UNC path if needed, for example:
\\server\share\%ENV_USERNAME%\Notes_Backup

8.3 Instructions – step by step

8.3.1 Zip installation package

It is recommended that the installation packages be zipped into a single 7-ZIP or ZIP archive once they have been created.

- [1]** Highlight all installation package folders and add them to an archive (in the following example, this is an IBM Notes 8.5.2 Client (English "only") + Fix Pack 2 + German and French language packs).

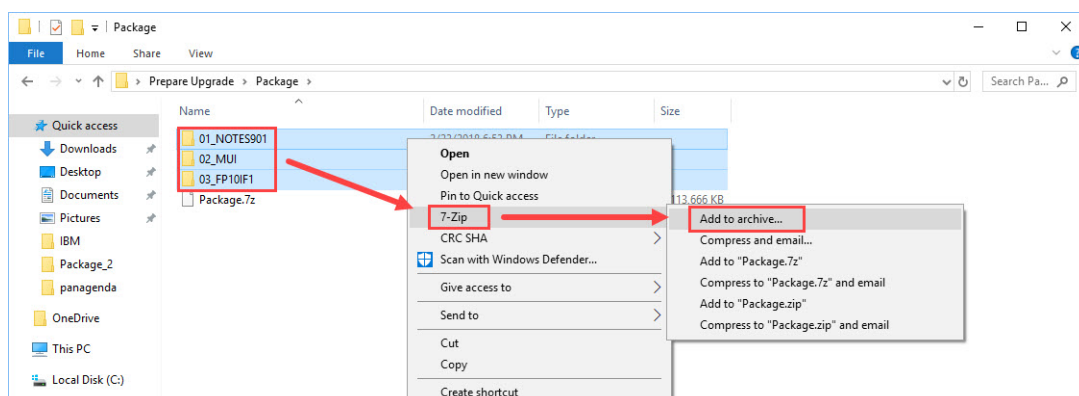


Figure 1: Installation Package – Add to archive...

- [2]** We recommend using 7-ZIP with *Compression level "Ultra"* since every byte counts when multiplied with the number of total users that are to download the upgrade package – whilst the zip process may take (very) long with "Ultra" settings, unzipping is pretty much as fast as if using normal compression:

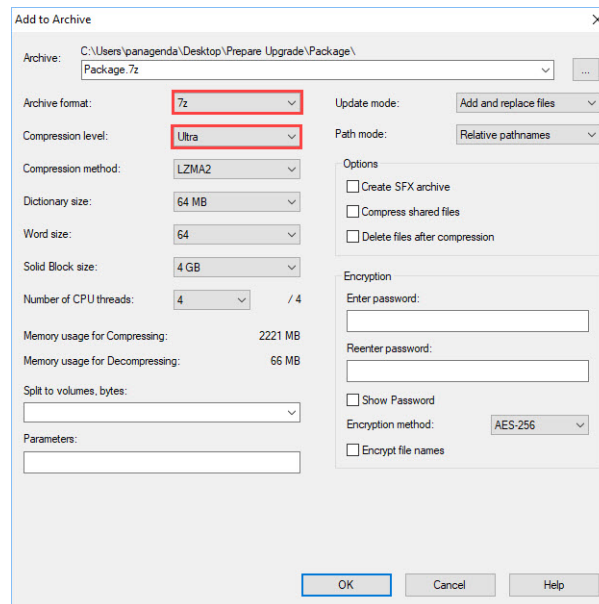


Figure 2: Installation Package – Archive Settings

8.3.2 MCKeyfileEditor.exe

The MCKeyfileEditor is used to store your Administrator Username(s) and Password(s) in a securely encrypted MCUUpgrade.mcu keyfile.

Any information entered into your keyfile is protected with a password of your choice for reading and editing the keyfile after saving it, as well as multi-level AES-256 bit encrypted.

You can enter multiple usernames and passwords in case your environment does not just have one administrator account for all computers.

Usernames, passwords and domains can be used in pairs or permuted, as in either (Admin1-PW1, Admin2-PW2, Admin3-PW3) for each domain, or (Admin1, Admin2 and Admin3) for each password and domain. Permutation is turned ON by default, and can be turned OFF by setting *PermutedLogon=0* in *MCUpgrade.ini*.



NOTE that if you do not supply the same number of admin usernames and passwords, permutation is forced ON.

Also, don't forget to supply the SMTP Username and Password for upgrade status emails.

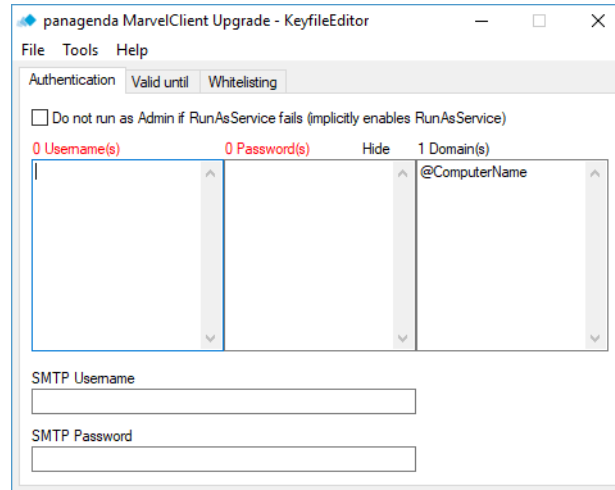


Figure 3: KeyfileEditor – Authentication

Note that you may want to change @ComputerName to YourDomainName (obviously adjusting YourDomainName as needed).



The permuted logon feature can be deactivated in MCUpgrade.ini (PermutatedLogon=0), if desired. When deactivated, the logon will be tried with admin username and password pairs – which requires the number of added usernames and passwords to match. In case multiple domains are supplied in the keyfile, these will always be permuted with the admin username and password pairs.

It is advisable to also supply SMTP authentication details in the keyfile for upgrade status emails. The authentication information should be usable from all respective clients to send report emails.

The use of SSL and port settings can be modified via MCUpgrade.ini (see "Status Report Emails and parameters in MCUpgrade.ini" on page 23).

8.3.3 MCUpgrade.exe and MCUpgrade.ini

MCUpgrade.exe contains the actual logic of MarvelClient Upgrade. Whether MC Upgrade is to perform an upgrade, install/uninstall, update etc. is controlled through the MCUpgrade.ini file, which needs to be prepared in itself and is later on generically personalized for each user via the MC Configuration database and DLL.

8.3.4 Visualizing Progress

Upon execution of your upgrade/update/install/uninstall, MCUpgrade visualizes progress as follows:

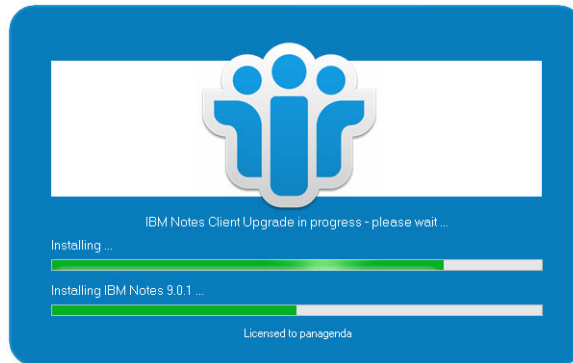


Figure 4: Upgrade Progress Visualization

Adjusting Progress Visualization

The visualization of the MC Upgrade progress can be individually customized. In order to do this, proceed as follows:

1. From a command line, run
MCUpgrade.exe /pbar
This unpacks a file named `panagenda_Splash.png` in the same folder.
2. In order to end the `MCUpgrade.exe` process, it needs to be killed e.g. via Windows Task Manager. When doing so, the "Please confirm" prompt of Windows Task Manager may be covered by the splash screen – if this is the case, simply press enter to confirm or drag the splash screen to a different location on the screen.
3. You can customize the `panagenda_Splash.png` file in an image editor. In this case, please pay attention to the position of text and progress bar, which are matched to the original file



Note: Once there is a file called `panagenda_Splash.png` in the same folder as `MCUpgrade.exe`, it will be used for visualization. The file should be customized in a program that supports transparency, aka **NOT Microsoft Paint**.

8.3.5 Parsing install or uninstall logs for errors

Due to MarvelClient's flexibility, it is possible to run any number and types of executables/programs during the upgrade. When combining multiple installers into a sequence of installs, it is often advisable to detect for whether a particular installer fails.

The following *MCUpgrade.ini* entry allows you to instruct MC Upgrade to parse install logs for errors:

- *InstallExecErrorSearchRegExp001=*

The "001" in the example above means that the log file defined under *InstallExec001=*, is searched for a particular (regular expression) string. As a prerequisite a log file must be created for the defined installation or uninstallation, as shown in the following example:

- *InstallExec001=msiexec /i "%MC_UPGRADEDIRECTORY%\901\IBM Notes 9.0.1 Social Edition.msi" /qb! /I*v "%MC_UPGRADEDIRECTORY%\msiinst.log"*



We recommend searching for the string „*.*(?![0-9]): 1603.**“:

- *InstallExecErrorSearchRegExp001=.*(?![0-9]): 1603.**

"Error Status: 1603" (attention, the string part "Error Status" depends on OS language and should hence not be searched for) is a common installation error indicator for IBM Software.

If the defined string is found, MarvelClient assumes that an error has occurred.



Note: *MarvelClient considers finding this string an error, even if no "real" installation errors have occurred.*

If, for example, you only search for „^. *1603.*\$*“, any line in the log file could trigger an "error" and MC Upgrade would cancel the installation process.*



Also Note: *If you install multiple packages (Feature Packs, Fix Packs, Connections Plugin etc.), you need to look in all log files for error strings by using as many *InstallExecErrorSearchRegExp###* entries as appropriate.*

*The same applies to errors when uninstalling software: *UninstallExecErrorSearchRegExp001*, *UninstallExecErrorSearchRegExp002*, etc.*

If an error occurs during an upgrade, the user receives an error message that s/he must confirm. Note that any timeout set for messages is ignored for the display of error messages. Any such errors are also reported back via Status Report Emails.

8.3.6 Status Report Emails and parameters in *MCUpgrade.ini*

Status Report Emails inform you whether or not an upgrade was successfully carried out. MC Upgrade requires specific SMTP information in order to send these. SMTP authentication details are stored in the MCUpgrade keyfile (also see “MCKeyfileEditor.exe” on page 19). All other settings can be set via *MCUpgrade.ini*:

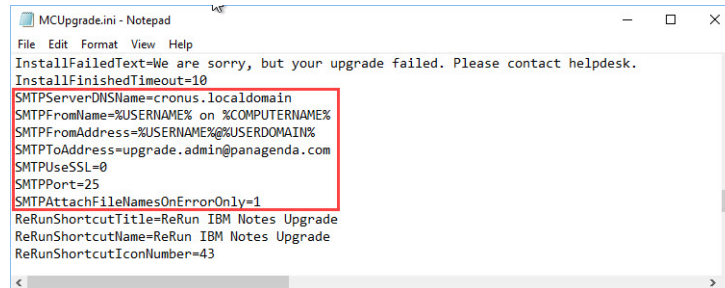


Figure 5: *MCUpgrade.ini* – SMTP Settings

SMTPAttachFileNamesOnErrorOnly=1 means that Windows Installer logs will only be attached to Status Report Emails in the case of errors. If you set the parameter to 0 or omit it (=default), log files will always be attached to all Status Report Emails.



Note: We recommend to always define a sender! Sender and receiver need to be valid SMPT addresses. In case you use a Mail-In database, the Mail-In document in the Domino directory must have an SMTP address.

8.3.7 Unzipping of compressed archives before uninstalling and installing

And 7z/zip compressed archives are unzipped before uninstalling and installing through the following MCUpgrade.ini entry:

UnzipArchive###=filename.7z

whereby ### must be a consistently ascending number, beginning with 001.

In addition, UnzipArchive### instructions allow for the following optional settings:

- **UnzipText###=Text** to display for progress visualization
- **UnzipCommand###=Option** to override the default “x” extraction command
- **UnzipSwitches###=Option** to override the default “-aoa -y” extraction switches
- **UnzipAppendix###=Option** to append further instructions to the unzip commandline resulting from above settings

- **UnzipArchiveToSubfolder###=1**

By default this option is set to 0, resulting in any 7z/zip file to be extracted into the same folder in which the 7z/zip file resides. If set to 1, the corresponding 7z/zip file is unzipped into a subfolder with the same name as the 7z/zip file.

- **UnzipDelete###=0**

By default, any 7z/zip file that is extracted via UnzipArchive### is automatically deleted after being unzipped to free up disk space. This behavior can be changed by setting the corresponding UnzipDelete### option to 0.

8.3.8 Uninstalling old IBM Notes client releases

To fully facilitate the generic concept of MC Upgrade, it is recommended that an existing IBM Notes client is always uninstalled and re-installed during a major release upgrade. There are two major reasons for this: a.) uninstall & install is about 1-2 minutes faster than an in-place upgrade and b.) in-place upgrades are by far not as successful as uninstall/install.



Note that for Feature Pack/Fix Pack updates and similar, existing clients must NOT be uninstalled.

Whilst uninstalling specific versions of IBM Notes is perfectly possible by using its so called Globally Unique Identifier (GUID) together with "msiexec /x", the much simpler, more robust and elegant solution to uninstalling any previous releases is the use of "NICE" (Notes Install Cleanup Executable):

<https://www-01.ibm.com/support/docview.wss?uid=swg21459714>

To do so, simply download *nice.exe* from above URL, deploy it along with your MCUpgrade components (*MCUpgrade.exe*, license file, keyfile, ...) and configure a corresponding Uninstall Instruction as follows in *MCUpgrade.ini*:

UninstallText001=Uninstalling old Notes client ...

UninstallExec001="%MC_MYUPGRADEDIRECTORY%\nice.exe" -rp -wipe -qn



Note that you may have to adjust the ...001 in case you have uninstall instructions you want to run before running NICE.

8.3.9 Installing a new IBM Notes client release

New IBM Notes client releases, Fix Packs, Feature Packs, Hotfixes, language packs and so on are installed through entries as highlighted in the following screenshot of *MCUpgrade.ini*:



```

MCUpgrade.ini - Notepad
File Edit Format View Help
ReRunShortcutName=ReRun IBM Notes Upgrade
ReRunShortcutIconNumber=43
NotesDataDirectoryOldExclude=*.ntf|*.log|*.nsf|*.dsk|*.cache|.ndk
DisallowRun001=notes.exe
DisallowRun002=notes.exe
DisallowRun003=admin.exe
DisallowRun004=designer.exe
DeferredFileMustExist001=C:\Users\Public\MCUpgrade\package.7z
DeferredFileMustExist002=C:\Users\Public\MCUpgrade\MCUpgrade.mcu
DeferredFileMustExist003=C:\Users\Public\MCUpgrade\license_upgrade.xml
NotesDataDirectoryOld=C:\Lotus\Notes\Data
NotesDataDirectoryNew=C:\Users\panagenda\AppData\Local\IBM\Notes\Data
NotesProgramDirectoryOld=C:\Lotus\Notes\
NotesProgramDirectoryNew=program_files_directory\IBM\Notes
MultiUserSharedDataNew=C:\ProgramData\IBM\Notes\Data\shared
UnzipArchive001=Package.7z
UninstallText001=Uninstalling IBM Lotus Notes ...
InstallExec001="C:\Users\Public\MCUpgrade\nice.exe" -rp -wipe -qn
InstallText001=Installing IBM Notes 9.0.1 ...
InstallExec001=msiexec /i "%MC_MYUPGRADEDIRECTORY%\01_Notes901\IBM Notes 9.0.1 Social Edition.msi" /qn /lme "%MC_MYUPGRADEDIRECTORY%\msiinst_Notes901.log" TRANSFORMS="%MC_MYUPGRADEDIRECTORY%\01_Notes901\notes901.mst"
InstallExecErrorSearchRegExp001=^.*: 1603\.$
InstallText002=Installing IBM Notes 9.0.1 Feature Pack 10 Interim Fix 1...
InstallExec002="%MC_MYUPGRADEDIRECTORY%\02_Notes901_FP10_IF1\clientupdate.exe" /s /u" /qn /lme "%MC_MYUPGRADEDIRECTORY%\msiinst_Notes901_FP10_IF1.log"
InstallExecErrorSearchRegExp002=^.*: 1603\.$
InstallFinishedExec=C:\Program Files (x86)\IBM\Notes\notes.exe
  
```

Figure 6: *MCUpgrade.ini* – Install Section

In the above example, the *InstallExec001* instruction leads to the execution of an *IBM Notes 9.0.1 Social Edition.msi* file, together with a corresponding transform file and various parameters, all from the folder *%MC_UPGRADEDIRECTORY%\01_NOTES901*, into which the necessary setup components were extracted.

As with most other *MCUpgrade.ini* entries, this setting, too, supports 0 to 999 entries, all of which must be consistently increased and, if used, begin with 001. This allows specify as many installs/uninstalls/parameters as needed for each upgrade/update case. It is imperative that the sequence of installations is configured in a sense-making way; for example, before installing a Feature Pack/Fix Pack, an IBM Notes client needs to be installed first.

8.3.10 Re-run of unsuccessful upgrades/updates

With the help of Status Report Emails, administrators know exactly in case and where upgrades were unsuccessful. As soon as the report – including attached log files – is received, administrators can begin with determining the cause and therefore the solution to the unsuccessful upgrade/update.

When the cause for an unsuccessful upgrade/update is determined and resolved, the upgrade process can be restarted on the affected machine. For ease of use and enhanced re-

run capabilities, MarvelClient Upgrade can automatically create a shortcut on a users' desktop in case of an error, all via the following configuration options:

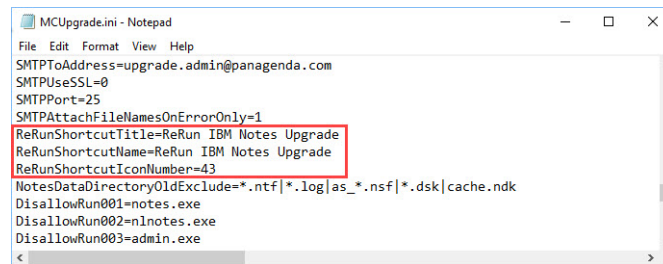


Figure 7: MCUpgrade.ini – ReRun Section

Note that icon settings are based on the Shell32 icons, see http://help4windows.com/windows_7_shell32_dll.shtml.

During a re-run *MCUpgrade.exe* are re-initiates according to the settings in *MCUpgrade.ini*. The shortcut created by MarvelClient Upgrade may also include a switch `"/ignoredeferralfilemustexist"`, then skipping any `DeferralFileMustExist###` settings. If desired, specific install and uninstall instructions can be skipped along, too: These can be defined via ODI (On Deferral Ignore) parameters:

- `InstallSkipODI00###`
- `UnInstallSkipODI00###`

If these parameters are set to "True" (=1), the corresponding install or uninstall parameters will be skipped during a re-run, too. For example:

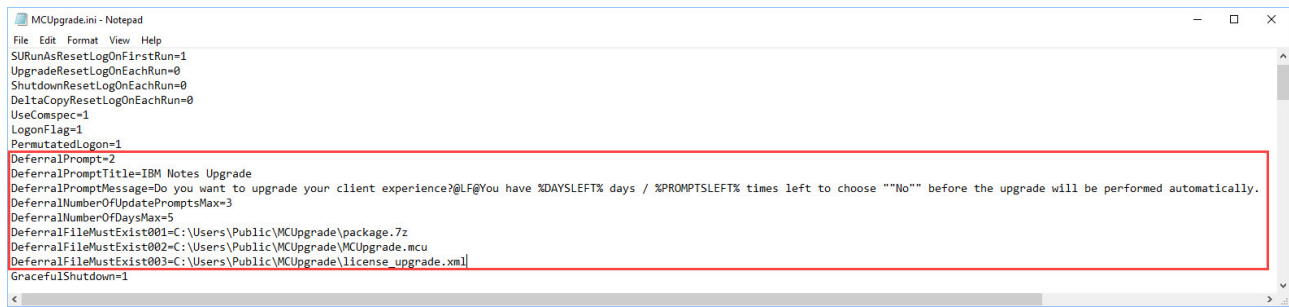
- `UninstallExec003=...`
`UnInstallSkipODI003=1`

8.3.11 Allowing users to defer an upgrade/update

The deferral section in *MCUpgrade.ini* can be modified to permit end-users to:

- postpone the upgrade of their IBM Notes client for a defined number of times
- postpone the upgrade of their IBM Notes client for a defined number of days

As soon as the user exceeds the specified number of deferrals OR if the number of specified days specified has passed after the date of the first deferral, the upgrade/update is enforced.



```

MCUpgrade.ini - Notepad
File Edit Format View Help
SUNRunAsResetLogOnFirstRun=1
UpgradeResetLogOnEachRun=0
ShutdownResetLogOnEachRun=0
DeltaCopyResetLogOnEachRun=0
UseComspec=1
LogonFlag=1
PermutatedLogon=1
DeferralPrompt=2
DeferralPromptTitle=IBM Notes Upgrade
DeferralPromptMessage=Do you want to upgrade your client experience?@LF@You have %DAYSLEFT% days / %PROMPTSLEFT% times left to choose "No" before the upgrade will be performed automatically.
DeferralNumberOfUpdatePromptsMax=3
DeferralNumberOfDaysMax=5
DeferralFileMustExist001=C:\Users\Public\MCUpgrade\package.7z
DeferralFileMustExist002=C:\Users\Public\MCUpgrade\MCUpgrade.mcu
DeferralFileMustExist003=C:\Users\Public\MCUpgrade\license_upgrade.xml
GracefulShutdown=1
  
```

Figure 8: MCUpgrade.ini – Deferral Section

The following parameters in *MCUpgrade.ini* allow you to control upgrade/update deferrals:

- [1] DeferralPrompt=2** – describes how often a user should be prompted for whether s/he wants an upgrade/update to be carried out: **0=never**, **1=once a day**, **2=always** (every time *MCUpgrade.exe* is executed. This typically happens during the client session using a MarvelClient Run Program Action with the Background Run Type).
- [2] DeferralPromptTitle=Upgrade?** – describes the title being displayed in the upgrade prompt window; defaults to "Upgrade" without quotes.
- [3] DeferralPromptMessage=Do you want to upgrade your client experience?@LF@You have %DAYSLEFT% days / %PROMPTSLEFT% times left to choose "No" before the upgrade will be performed automatically.**
 Defaults to above message and allows for customizing the message being displayed to the end users in the upgrade prompt window. Use "@LF@" anywhere in the text to create a new line/vertical spacing. The variables **%DAYSLEFT%** and **%PROMPTSLEFT%** can be used to display the remaining days and/or prompts for each user (optional).
- [4] DeferralNumberOfUpdatePromptsMax=3** – describes how many times a user can postpone an upgrade, independent of how many days may still be left according to the next parameter; defaults to 0 = unlimited.
- [5] DeferralNumberOfDaysMax=5** – describes the number of days since the very first prompt within which the upgrade must be run in any event; independent of how many times a user has been asked. Set to 0 for infinite number of days

- [6] DeferralFileMustExist###=%TEMP%\MCUpgrade.exe** – Check for up to 999 files that must be present before the deferral prompt actually appears. This is to avoid executing an Upgrade before all necessary software is readily deployed. Supports usage of OS environment variables, such as %TEMP%, %LOCALAPPDATA% and similar.

8.3.12 Graceful Shutdown

The Graceful Shutdown feature enables you to initialize an Upgrade/Update whilst a client is up and running (via the Background RunType in the Configuration database). End users may have unsaved documents, where killing the Notes client would lead to data losses. With the GracefulShutdown options in *MCUpgrade.ini* you can modify the details about the behavior and appearance of this feature:

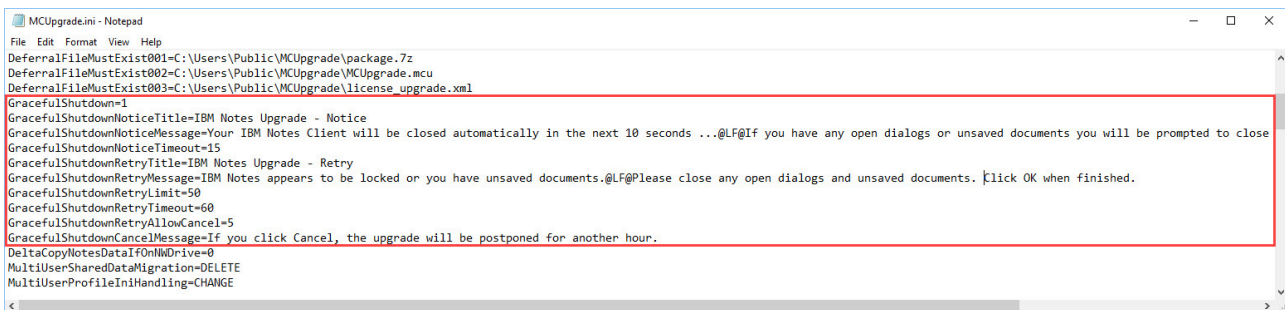


Figure 9: *MCUpgrade.ini* – Graceful Shutdown Section

- [1] Graceful Shutdown=1** – this parameter is used to activate the Graceful Shutdown feature: **0=disabled** (client is killed, data loss possible), **1=enabled (default)**, **2=enabled with fallback to immediate client kill**

- [2] GracefulShutdownNoticeTitle=** and **GracefulShutdownNoticeMessage=**

Per default there is no initial shutdown notice; Example settings for enabling:

- *GracefulShutdownNoticeTitle=IBM Notes Client Upgrade – Notice*
- *GracefulShutdownNoticeMessage=Your IBM Notes Client will be closed automatically in the next %1 seconds ...@LF@If you have any open dialogs or unsaved documents you will be prompted to close them so that the client upgrade can continue.*

GracefulShutdownNoticeTimeout=60 – Only relevant if both *GracefulShutdownNoticeTitle* and *GracefulShutdownNoticeMessage* are set. Defaults to 60 and defines the number of seconds before the first client shutdown is attempted. This number replaces %1 in *GracefulShutdownNoticeMessage*.

[3] **GracefulShutdownRetryTitle=** and **GracefulShutdownRetryMessage=** – wording of the retry message(s). Will only appear in case a shutdown attempt fails (because of open documents or dialogs). Defaults are:

- **GracefulShutdownRetryTitle=IBM Notes Client Upgrade - Retry**
- **GracefulShutdownRetryMessage=IBM Notes appears to be locked or you have unsaved documents.@LF@Please close any open dialogs and unsaved documents. Click OK when finished.**

GracefulShutdownRetryLimit=50 – defines how often MarvelClient Upgrade should try to shut down the client; defaults to 50.

GracefulShutdownRetryTimeout=60 – defines timeout between shutdown attempts; defaults to 30.

If you are running MCUUpgrade via MarvelClient background thread, timeout*limit may not exceed the background thread interval (defaults to one hour / 3,600 seconds).

GracefulShutdownRetryAllowCancel=5 – displays a cancel button after the 5th retry (according to this example); defaults to **0=never**.

GracefulShutdownCancelMessage=If you click Cancel, the upgrade will be postponed for another hour. Please modify this text in accordance with your execution interval (for the MarvelClient background thread this defaults to one hour).

8.3.13 Prepare upgrade components for distribution


As soon as the *MCUpgrade.mcu* and *MCUpgrade.ini* are readily prepared, they should be zipped into one joint archive, together with *MCUpgrade.exe*, *MCUpgrade.dll*, *license_upgrade.xml* and *panagenda_Splash.png* (if you are using a customized splash screen), all of which can then be distributed by MarvelClient:

[1] Select all relevant upgrade files, right-click → **Send to** → **Compressed (zipped) folder**:



Note: You may want to add multiple *MCUpgrade.ini* files into the same zipfile and have the MarvelClient configuration later determine which upgrade path to execute for which users.

[2] Rename the resulting zip file to, for example, *MCUpgrade.zip*:

Name	Date modified	Type	Size
 MCUpgrade.zip	2/26/2018 5:36 PM	Compressed (zipp...	1,827 KB

[3] The resulting *MCUpgrade.zip* archive is referred to later in instructions further down.

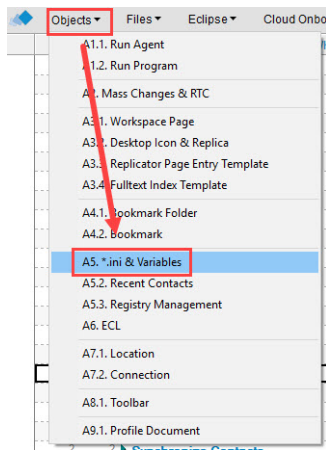
8.3.14 Smart File Downloader

The Smart File Downloader helps distribute software packages, using the MarvelClient extension in IBM Notes. You can skip this section, if you cannot or do not want to use Smart File Downloader.



Note that we do NOT recommend running installers directly from network drives! If you do not use Smart File Downloader, please make sure that you deploy the software packages onto each client with other tooling.

- [1] Smart File Downloader can be configured different from its default behavior by setting the following variables in *notes.ini*: a **Notes.ini & Variables** Action in the Configuration database makes this an easy task:



Note that for production purposes the Smart File Downloader does not need to be configured different from the recommended default.

The following two variables can be set:

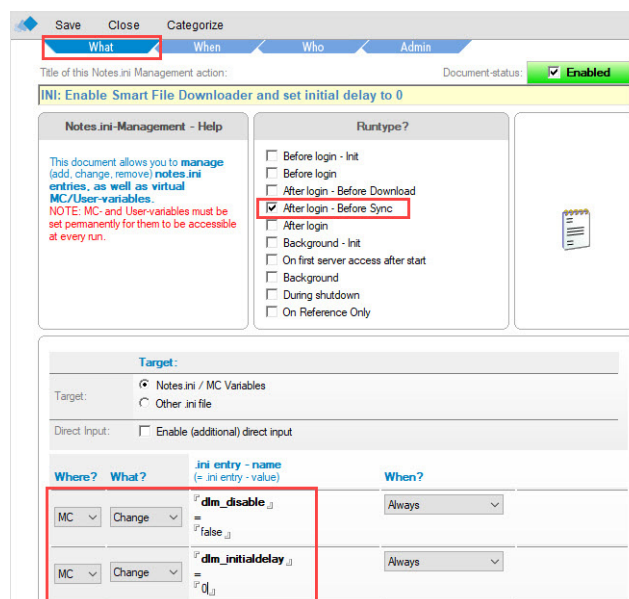


Figure 10: Smart File Downloader – Notes.ini Management

mc:dlm_disabled=true

OR

mc:dlm_disabled=false (default)

DISABLES/ENABLES Smart File Downloader on the client – since it is enabled by default, this is only documented here for reasons of completeness; setting mc:dlm_disabled to false as outlined in the following screenshot is NOT necessary.

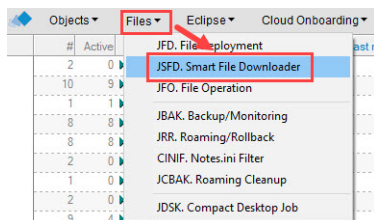
mc:dlm_initialdelay=<NumberOfMinutes>

delay in minutes for earliest initiation of the download after the client has been started; defaults to 5 minutes = a client will wait at least 5 minutes before even calculating further download procedures.



Note that a.) mc:dlm_disabled=false is not needed, b.) for testing it makes sense to set mc:dlm_initialdelay to 0. For production, however, you should either leave the default of 5 minutes or change it to no less than 2 minutes as to make sure that in case downloading starts (see further down on how Smart File Downloaded determines when to download how much), the user or replicator task are not disturbed.

[2] Create a Smart File Downloader document in the MC Configuration database:



[a] Select the file to be distributed from the file system using **Select Source File**.



Note that for performance reasons it is highly advisable to select a LOCAL source file for chunking the download (see [d]). Also note that it is vital that the so LOCALLY selected file is the exact same (bitwise) as the one that users will download from the fileserver for hashing purposes (see [d]).

[b] Specify source filepath including filename where clients will be able to download the file. This can also be a UNC path (for example: \\ServerName\Path\File.exe)

[c] Specify the target directory in which the file is to be stored on the client (requires write permission by end users). This directory is needed later on in order to start an upgrade/update. Supports use of MC References.

- [d]** Set chunk size. A chunk = a virtual piece of a package. For instance, a 1 GB file can be virtually chunked into 10 x 100 MB chunks. Use **(Re)Compute Packets** to calculate the number of packets and unique hash values.



Note that the file is not physically chunked, but that only hashvalues are calculated for each chunk. In case downloading a chunk (=piece) fails, MarvelClient will only re-download the failed chunk instead of the whole package. Also note that in case downloading is interrupted, MarvelClient will continue were it left off last time – resume is bitwise, not just per chunk.

TIP



We've seen customers having trouble with calculating **chunk sizes larger than 10-20 MB** – if so, tune your jvm properties files as shown on slide 24 of the following presentation and/or adjust JavaMaxHeapSize in notes.ini (e.g., set JavaMaxHeapSize=256MB): <https://www.slideshare.net/panagenda/dev1185-ibm-notes-performance-boost-reloaded-ibm-connect-2017>

- [e]** Set time window: For instance, a maximum of five chunks of the file may be downloaded with a maximum bandwidth of 5 Mbits each day between 08:00 and 12:00 hrs. With a little bit of math (knowing how many users sit "behind" what kind of network connection (10 Mbit, 100 Mbit, 1Gbit or similar) allows you to ensure that your network isn't bloated "Monday morning at 8 am because all users want to download a large package all at once".

For testing purposes you may want to set the number of packages to 100,000 even if there are lesser chunks to download – the number of chunks is balanced in 1 minute intervals across the entire time window by pure randomness – this ensures that some users will start 10 minutes after client startup, others 2 hours later, and so on. With 100,000 chunks to be downloaded in an 8 hour time window, clients will download a chunk every minute. Depending on your bandwidth settings (and bandwidth availability at the time of downloading), a maximum chunks settings of 100,000 may well work in a GBit environment.



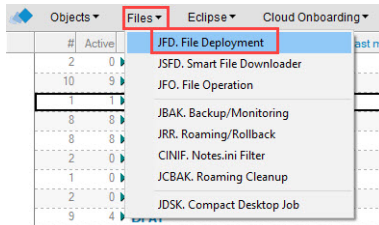
Note: Make sure that you set this configuration back to sense making values once you move into production!

- [f]** Restrict the Smart File Downloader Action to users, groups and/or Conditions to control which clients effectively download the file. Also also chapter "Useful MarvelClient Features for MC Upgrade" on page 38.

8.3.15 Distribution of upgrade files

The components zipped in chapter “Prepare upgrade components for distribution” on page 29 are now distributed to the clients by a **File Deployment** Action.

[1] Create a **File Deployment** Action:



[2] Add the previously created *MC_Upgrade.zip* file, enter the desired target directory and check the **Unzip** option. The local app data directory of the OS user initiating the upgrade is used in the following screenshot, however it is recommended to use a folder underneath Users\Public (e.g., <env:PUBLIC>\panagenda\MCUpgrade), so that multiple operating system users can progress with the same download on shared computers.

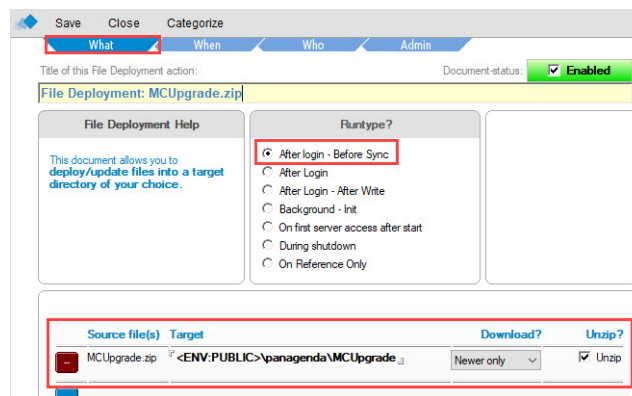


Figure 11: File Deployment – Add *MC_Upgrade.zip*



Note: The target directory selected here is the *%MC_UPGRADEDIRECTORY%* directory referred to in previous chapters.

[3] The Action should now be linked with adequate Conditions on the **When** tab. Select MarvelClient Conditions for MC Upgrade are recommended in the chapter “Useful MarvelClient Features for MC Upgrade” on page 38.

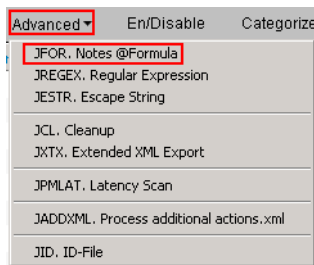
8.3.16 Language Packs and the *notes.ini* Region setting



Note: This step only needs to be carried out when installing language packs (whether as an update or as part of an upgrade)!

If a language pack is (also) installed, the region can be read from the previously/currently installed IBM Notes client. Thereby, the correct language parameter can be passed to a language pack install, ensuring enablement of the correct language for each user.

[1] Create a **Notes @Formula** Action:



[2] The following example identifies the client region from a client's *notes.ini*: The target language for the language pack will be set to French (FR) if the region was previously FR, and to German (DE) if it was previously DE. In all other cases the client language will be set to English (EN). The formula can be modified as needed.

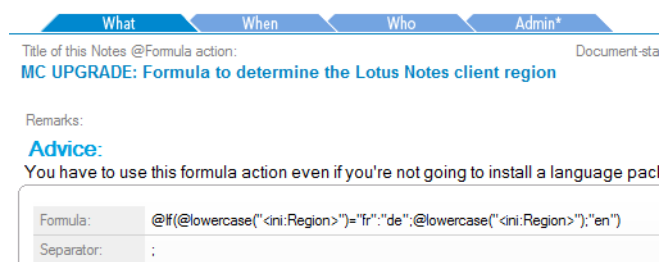


Figure 12: Notes @Formula – Set client region

[3] Formula in cleartext:

@If(@lowercase("<ini:Region>")="fr";"de";@lowercase("<ini:Region>");"en")

[4] **Important:** An alias must be assigned to the formula Action on the *Admin* tab so that the result of the formula can be queried from another Action. Recommended: "clientregion".

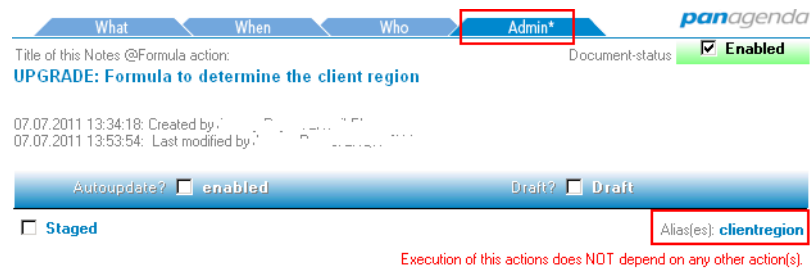


Figure 13: Notes @Formula – Alias

8.3.17 Generic customization of *MCUpgrade*.ini*

For the most a generic approach of client upgrades (=independent of where end users have their clients installed), *MCUpgrade.ini* is finalized with dynamic values for the old client directories, which are subsequently passed on to the installation process. With one single Action, all relevant **user- individual** parameters are written into *MCUpgrade.ini*.

The mandatory parameters for a MAJOR upgrade are the following:

1. Old IBM Notes data directory (dynamic)
2. New IBM Notes data directory
3. Old IBM Notes program directory (dynamic and calculated on the go).
4. New IBM Notes program directory
5. (OPTIONAL) Old client region for language pack installations (as previously identified by formula)

[1] Create a *.ini & Variables Action as follows:

- Set the Runtype to *During shutdown*
- Configure the *Filepath* of the *.ini file to point to the respective *MCUpgrade*.ini* file, in this example:
`<env:PUBLIC>\panagenda\MCUpgrade\MCUpgrade_NOTES901.ini`
- Configure *NotesDataDirectoryOld* and *NotesDataDirectoryNew* (only needed for when upgrading from one Notes release to another, not for Feature Packs/Fix Packs, Hotfixes or Addons):
`NotesDataDirectoryOld=<notes:data_directory>`
`NotesDataDirectoryNew=<os:local_app_data_directory>\IBM\Notes\Data`

- Configure `NotesProgramDirectoryOld` and `NotesProgramDirectoryNew` (only needed for when upgrading from one Notes release to another, not for Feature Packs/Fix Packs, Hotfixes or Addons):

`NotesProgramDirectoryOld=<notes:client_directory>`

`NotesProgramDirectoryNew=<os:program_files_directory>\IBM\Notes`

- Configure memorizing the user's current client language (only needed for when installing language packs)

`NotesLanguageOld=<clientregion:result=>`



Note that the equal sign at the end of result is intended and means "if the result of the Action with the alias "clientregion" cannot be resolved, then pass back an empty string"

- If needed, you can also configure OS environment variables that are to be set at the beginning of an upgrade - for example:

`SetEnvName001=MC_OS_PROGRAM_FILES_DIR`

`SetEnvValue001=<os:program_files_directory>`

Environment variables can be useful to generically work with varying directories across different versions of Microsoft Windows; for example, the program files directory is different on Windows 7/8/8.1/10 to WindowsXP or if you use 32Bit and 64Bit Clients. In above example `%MC_OS_PROGRAM_FILES_DIR%` can then be reused during an upgrade for pretty much any filepath and directory.

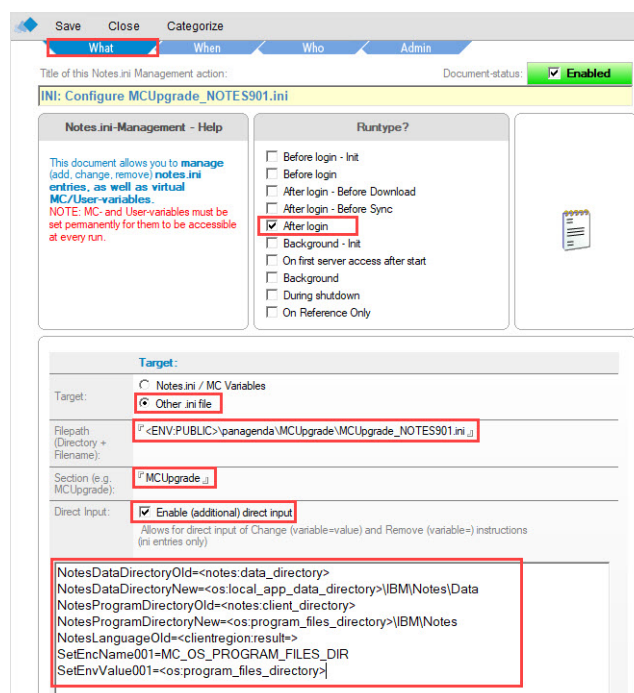


Figure 14: Notes.ini Management – Customize MCUpgrade*.ini



Note that *NotesLanguageOld* only needs to be set along with the *clientregion @formula* Action if you (also) install language packs!

Also note that the **Old* and **New* directories only need to be set for major client upgrades and especially when switching from single to multi-user client installs or vice versa.

[2] Note: Unless installed differently, the target data directory for a multi-user installation should be specified as follows: `<os:local_app_data>\IBM\Notes\Data`

[3] Note: Unless installed differently, the target program directory for a multi-user installation should be specified as follows: `<os:program_files_directory>\IBM\Notes`

[4] This ***.ini & Variables** Action should be linked with adequate Conditions. Select MarvelClient Conditions for MC Upgrade are recommended in the chapter "Useful MarvelClient Features for MC Upgrade" on page 38.

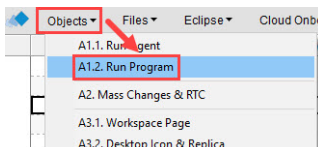
8.3.18 Launching MarvelClient Upgrade

The actual start of the an upgrade is ultimately triggered by a Run Program Action. It calls *MCUpgrade.exe* and tells it where to find "*MCUpgradeMain.exe*" and "*MCUpgrade.ini*".



NOTE: *MCUpgradeMain.exe* is included in *MCUpgrade.exe* and auto-deployed before switching to an administrator account.

[1] Create a Run Program Action



[2] Configure the Action as follows:

<input type="checkbox"/> Background <input checked="" type="checkbox"/> During shutdown <input type="checkbox"/> On Reference Only	
Program & (optional) parameter(s):	
Program path:	<input type="text" value="<os:local_app_data_directory>\panagenda\MCUpgrade\SURunas.exe"/> <small>e.g. winword.exe or <os:program_files>\Microsoft Office\Office12\winword.exe</small>
Parameters:	<input type="text" value='"<os:local_app_data_directory>\panagenda\MCUpgrade\MCUpgrade.exe'/> <input type="text" value='"<os:local_app_data_directory>\panagenda\MCUpgrade\MCUpgrade_LN853.ini'/>
Action:	Display:
<input type="radio"/> Launch Notes process <input checked="" type="radio"/> Open <input type="radio"/> Edit <input type="radio"/> Explore <input type="radio"/> Find <input type="radio"/> Print <input type="radio"/> Properties	Visibility: <input type="radio"/> Visible <input checked="" type="radio"/> Invisible
Wait?	<input checked="" type="radio"/> Do not wait for program to finish <input type="radio"/> Wait for program to finish

Figure 15: Run Program – Perform Update

The program path used above is:

<env:PUBLIC>\panagenda\MCUpgrade\MCUpgrade.exe

The parameter field is configured as follows (incl. quotes):

"<env:PUBLIC>\panagenda\MCUpgrade\MCUpgradeMain.exe"

"<env:PUBLIC>\panagenda\MCUpgrade\MCUpgrade_NOTES901.ini"



Note that the parameters are separated by a blank (space), NOT carriage return!

[3] This **Run Program** Action should be linked with useful Conditions. Select MarvelClient Conditions for MC Upgrade are recommended in the chapter "Useful MarvelClient Features for MC Upgrade" on page 38.

8.4 Useful MarvelClient Features for MC Upgrade

The MarvelClient Actions referred to in this manual should be made dependent on appropriate Conditions, e.g., to prevent unnecessary or unwanted execution. Select Conditions, that have proven to be of repeated value, are described below. The use of the Conditions is not explained in detail, assuming that customers are familiar with MarvelClient Conditions and References; if not, please refer to the MarvelClient Administrator's Guide (↗ "References" on page 35 and "When Tab in Detail" on page 109).

8.4.1 Querying IBM Notes client release (text-based)

Create a String Condition and check for the target client version – i.e. the client version **AFTER** the upgrade.

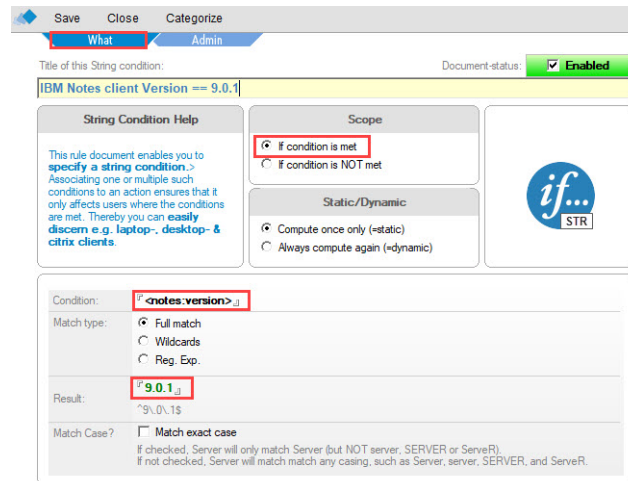


Figure 16: String Condition – Check Notes version

Useful in the following actions:

- Smart File Downloader
- Launching MarvelClient Upgrade
- Updating MCUpgrade*.ini



Note that using <notes:version_number> may be better in combination with a Numeric Condition, as it allows to reliably check for 9.0.1 FP10 and similar, which is impossible with <notes:version> (as notes:version does not contain Feature Pack/Fix Pack details).

8.4.2 Querying IBM Notes client release (numeric)

Create a Numeric Condition and check for the target client version – i.e. the client version **AFTER** the upgrade.

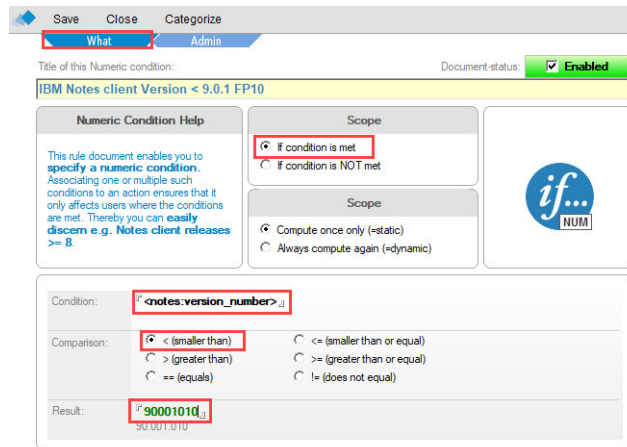


Figure 17: Numeric Condition – Check Notes version

Useful in the following actions:

- Smart File Downloader
- Launching MarvelClient Upgrade
- Updating *MCUpgrade*.ini*

8.4.3 Checking for sufficient disk space and memory

It is recommended that disk space and physical memory are checked before rolling out installation packages and starting installations. For instance, there is little point in installing IBM Notes 9 Standard on a computer with just 1 GB of RAM. Sufficient free disk space is obviously also necessary for deployment and installation. MarvelClient offers readily prepared example Conditions in the **Configuration database** that can be used as is or adapted as needed:










			Free Disk Space in Data > 250	[<os:disk_data_free_mb>] >= [250]
			Free Disk Space in Program > 750 MB	[<os:disk_client_free_mb>] >= [750]
			Memory > 1 GB / 1000 MB	[<os:memory_physical_total_mb>] >= [1000]

Figure 18: Example Conditions – Diskspace/RAM

Useful in the following actions:

- Smart File Downloader
- Launching MarvelClient Upgrade

8.4.4 Cleanup of *notes.ini* after an upgrade

If applicable, *notes.ini* entries, that are no longer required after an upgrade, should be removed as soon as the upgrade has completed successfully. Such an Action can be made dependent on several Conditions such as IBM Notes and Feature Packs/Fix Pack version, and similar.

8.4.5 Deleting installation packages after an upgrade

After successful installation, any therefor deployed installation packages should be deleted to free up disk space. While by default, 7-ZIP/ZIP archives are deleted automatically after decompression, files and directories unzipped from therein remain on disk. A MarvelClient **Run Program** Action can be used to execute a delete command at operating system level. An Example Action titled ***"Delete all files in os:temp_directory (including files in subdirectories)"*** is located in the Configuration database, *"All Actions > Examples"* view, and can be adapted to point to the upgrade directory used in the above chapters (e.g., `<env:PUBLIC>\panagenda\MCUpgrade`).

8.5 Performing an update only

The following screenshot shows an excerpt of an *MCUpgrade.ini* – further referred to as *MCUpdate.ini* – used for 'just' an update, where no client upgrade (uninstall, migrate data directories, install, etc.) as such is performed, but only an update such as installing a Feature Pack/Fix Pack or a newer Sametime embedded version, Connections plugins and similar. Hence, various parameters do not need to be specified, such as Old and New Program and Data directory, and many more:

The following screenshot shows that an *MCUpgrade.ini* is much shorter than a full-blown *MCUpgrade.ini*:

```
[MCUpgrade]
SIRunAsResetLogOnFirstRun=1
UpgradeResetLogOnEachRun=0
ShutdownResetLogOnEachRun=0
DeltaCopyResetLogOnEachRun=0
UseComspec=1
LogonFlag=1
PermutatedLogon=1
DeferralPrompt=2
DeferralPromptTitle=IBM Notes Update
DeferralPromptMessage=Do you want to update your client experience?@LF@You have %DAYSLEFT% days / %PROMPTSLEFT% times left to choose ""No"" before the update will be performed automatically.
DeferralNumberOfUpdatePromptsMax=3
DeferralNumberOfDaysMax=5
DeferralFileMustExist001=C:\Users\Public\MCUpgrade\package.7z
DeferralFileMustExist002=C:\Users\Public\MCUpgrade\MCUpgrade.mcu
DeferralFileMustExist003=C:\Users\Public\MCUpgrade\license_upgrade.xml
GracefulShutdown=1
GracefulShutdownNoticeTitle=IBM Notes Update - Notice
GracefulShutdownNoticeMessage=Your IBM Notes Client will be closed automatically in the next 10 seconds ...@LF@If you have any open dialogs or unsaved documents you will be prompted to close them so that the cl
GracefulShutdownNoticeTimeout=15
GracefulShutdownRetryTitle=IBM Notes Update - Retry
GracefulShutdownRetryMessage=IBM Notes appears to be locked or you have unsaved documents.@LF@Please close any open dialogs and unsaved documents. Click OK when finished.
GracefulShutdownRetryLimit=50
GracefulShutdownRetryTimeout=60
GracefulShutdownRetryAllowCancel=5
GracefulShutdownCancelMessage=If you click Cancel, the update will be postponed for another hour.
DeltaCopyNotesDataIfOnNWDrive=0
SkipNotesDataDirectoryMigration=1
DeleteNotesDataDirectory01dIfOnNWDrive=0
DeleteNotesINI=0
StopService001=Lotus Notes Diagnostics
StopService002=Multi-user Cleanup Service
StopService003=Lotus Notes Single Logon
StopService004=LNSUSvc
StopService005=uuaserv
StartService001=uuaserv
KillProcessWaitTime=15
KillProcess001=notes.exe
KillProcess002=nlnotes.exe
KillProcess003=notes2.exe
KillProcess004=ntaskldr.exe
KillProcess005=nminder.exe
KillProcess006=designer.exe
KillProcess007=admin.exe
KillProcess008=nhttp.exe
KillProcess009=sametime75.exe
KillProcess010=nminder.exe
KillProcess011=soffice.bin
KillProcess012=soffice.exe
KillProcess013=nssd.exe
KillProcess014=notesmm.exe
IsDebuggingEnabled=0
IsSimulateEnabled=0
RestoreNetworkDriveMappings=0
ProgressBarTitle=IBM Notes Update
ProgressBarText=IBM Notes Client Update in progress - please wait ...
InstallFinishedTitle=IBM Notes Update
InstallFinishedText=Update done successfully. Thank you for your patience.
InstallFailedTitle=IBM Notes Update
InstallFailedText=We are sorry, but your update failed. Please contact helpdesk.
InstallFinishedTimeout=10
SMTPServerDNSName=cronus.localdomain
SMTPFromName=%USERNAME% on %COMPUTERNAME%
SMTPFromAddress=%USERNAME%@%USERDOMAIN%
SMTPToAddress=upgrade.admin@panagenda.com
SMTPUseSSL=0
SMTPPort=25
SMTPAttachFileNamesOnErrorOnly=1
ReRunShortcutTitle=ReRun IBM Notes Upgrade
ReRunShortcutName=ReRun IBM Notes Upgrade
ReRunShortcutIconNumber=43
DisallowRun001=notes.exe
DisallowRun002=nlnotes.exe
DisallowRun003=admin.exe
DisallowRun004=designer.exe
UnzipArchive001=Package.7z
InstallText001=Installing IBM Notes 9.0.1 Feature Pack 10 Interim Fix 1...
InstallExec001="%MC_MYUPGRADEDIRECTORY%\02_Notes901_FP10_IF1\clientupdate.exe" /s /v"/qn /lme "%MC_MYUPGRADEDIRECTORY%\msinst_Notes901FP10IF1.log""
InstallExecErrorSearchRegExp001="^.*: 1603\.$
InstallFinishedExec=C:\Program Files (x86)\IBM\Notes\notes.exe
```

Figure 19: *MCUpgrade.ini*

Terminating processes may still be relevant for just an update, like installing a Feature Pack/Fix Pack or updating Sametime Embedded – all in all you can configure your own *MCUpgrade_or_Update.ini* file as you see fit.



Note that in any case you should NOT terminate services if you only choose to perform an update, like installing a Feature Pack/Fix Pack, as the services would not be started again thereafter. Also, a Feature Pack/Fix Pack install does not require services to be terminated (as opposed to a complete client upgrade).

8.6 Description of *MCUpgrade.ini* entries

<i>MCUpgrade.ini</i> entries	
Entry (=default, if any)	Description
<i>SURunAsResetLogOnFirstRun=1</i>	Resets the log file upon every run of <i>MCUpgrade.exe</i> ; recommended to not set / set to 1
<i>UpgradeResetLogOnEachRun=0</i> / not set	Resets the logfile when switching from <i>MCUpgrade.exe</i> to <i>MCUpgradeMain.exe</i> ; not recommended to ensure complete logging
<i>UpgradeDirectory=not set=folder in which MCUpgrade.exe resides</i>	What to set <i>%MC_UPGRADEDIRECTORY%</i> (=OS environment variable) to, which is used by i.e. 7-Zip self-extractor, and other components of <i>MCUpgrade</i>
<i>NotesDataDirectoryOld</i>	Mandatory for client upgrades to detect whether the old client had a different data directory and if so copy/move files into new client
<i>NotesDataDirectoryNew</i>	Similar to <i>NotesDataDirectoryOld</i> , but target folder
<i>NotesProgramDirectoryOld</i>	Mandatory for client upgrades to detect whether the old client had a different data directory and if so copy/move files into new client and also optionally adjusts <i>notes.ini</i> entries across multi-user data directories
<i>NotesProgramDirectoryNew</i>	Similar to <i>NotesProgramDirectoryOld</i> , but target folder
<i>DeleteNotesDataDirectoryOldIfOnNWDrive=0</i>	If the old data directory was on a network drive, will delete the entire directory after uninstalling the old client
<i>DeleteNotesINI=1</i>	If set to 1, will delete <i>notes.ini</i> (in both old data and program directory) before installing the new client; this is to prevent cases where the installer installs the new client into the same locations as the old client instead of into standard new paths. Defaults to 1 if not set and no (0) <i>UninstallExec*</i> instructions are configured. Defaults to 0 if not set and one or more <i>UninstallExec*</i> instructions are configured.
<i>RemoveShortCutsOnStart001-999</i>	0 to 999 regular expressions <i>MCUpgrade</i> uses to search for and delete shortcuts. Shortcuts are searched for on the Desktop and in the Start Menu of all users, the Admin user and the user who initiated <i>MCUpgrade</i> . Example.: <i>RemoveShortCutsOnStart001=^.*\\notes.exe\$</i>
<i>RemoveShortCutsAlsoInDirectory001-999</i>	0 to 999 directories where <i>MCUpgrade</i> should also search for shortcuts in addition to the standard user desktop, all user desktop and such

Table 2: *MCUpgrade.ini* Entries

MCUpgrade.ini entries	
Entry (=default, if any)	Description
PreUninstallRemoveRegistryKeyName001-999 together with PreUninstallRemoveRegistryKeyValue001-999	Allows to remove 0-999 specific registry values or entire keys prior to uninstalling software (if PreUninstallRemoveRegistryKeyValue### is set to * (asterisk), the entire key as specified in the corresponding PreUninstallRemoveRegistryKeyName is deleted)
PreInstallRemoveRegistryKeyName001-999 together with PreInstallRemoveRegistryKeyValue001-999	Same as above but prior to installing software (= after uninstall)
StopService001-999	0 to 999 services that should be stopped before uninstall/install
ExitProcessMaxRuntimeMinutes001-999	Max. number of minutes a process (see next entry) is allowed to run. If runtime of a process is <= MaxRuntime then MarvelClient Upgrade will exit. If runtime of a process is > MaxRuntime, then MarvelClient will kill it.
ExitProcess001-999	0-999 process names that are checked for MaxRuntime (see above) – e.g., msixec or software deployment tooling
KillProcessWaitTime=60 / not set	Number of seconds to wait until the following processes are killed
KillProcess001-999	0-999 process names to kill after KillProcessWaitTime has passed and a process is still running
IsDebuggingEnabled=0 / not set	If set to 1 will display (many) message boxes during an MC Upgrade run to follow through the progress step by step for testing purposes
IsSimulateEnabled=0 / not set	If set to 1 will only log what would have happened but effectively not move/delete data, uninstall, install etc. Don't forget to set this to 0 for testing/production
RestoreNetworkDriveMappings=0	When switching from a user account to the admin account, any previous network drive mappings are lost; if this variable is set to 1, MarvelClient Upgrade will memorize any previously mapped network drives and attempt to remap them after having switched to admin. This is vital in cases where the data directory is and shall remain on a network drive! We strongly recommend to move IBM Notes client data directories from network drives to local fixed disks for performance reasons. This can easily be done by setting DeltaCopyNotesDataIfOnNWDrive=1 in MCUpgrade.ini together with a normal multi-user upgrade and by setting NotesDataDirectoryOld and ...New accordingly.
LogFile=not set=MCUpgrade.exe folder + MCUpgrade.log	Points to where the MC Upgrade log file is to be maintained; full path required = folder AND filename.
ProgressBarTitle (deprecated when using MCUpgradeProgress.exe)	Allows for customization of the upgrade/update progress visualization

Table 2: MCUpgrade.ini Entries

MCUpgrade.ini entries	
Entry (=default, if any)	Description
<i>ProgressBarText=note set=Upgrade in progress - please wait...</i>	Allows to change text above progress bars
<i>ProgressBar1StaticText</i>	Allows for a STATIC label of first progress bar
<i>ProgressBar2StaticText</i>	Allows for a STATIC label of second progress bar

Table 2: MCUpgrade.ini Entries

8.7 Series of events within MCUpgrade(Main).exe

MCUpgrade.exe executes whatever is configured in the respective MCUpgrade.ini in the following order – ALL as the user who invoked the upgrade:

1. 1.Initialize configuration
2. Open log file
3. Check keyfile and license
4. Check for parallel run of MCUpgrade
5. Check for ExitProcess###
6. Check for DeferralFileMustExist
7. Check for DeferralAdminMustExist
8. DeferralPrompt
9. GracefulShutdown
10. Initialize progress visualization
11. Memorize Network Drive Mappings (if so configured and applicable)
12. Switch to admin account including UAC elevation → MCUpgradeMain.exe

MCUpgradeMain.exe executes whatever is configured in the respective *MCUpgrade.ini* in the following order – ALL using an administrator account:

1. Initialize configuration
2. Open log file
3. Initialize progress visualization
4. Check license
5. Set OS Environment variables (if so configured)
6. Apply DisallowRun/RestrictRun
7. Removing shortcuts (if so configured)
8. Restore Network Drive Mappings (if so configured and applicable)
9. Stop Services (if so configured)
10. Checking processes (if so configured)
11. Terminating processes (if so configured and applicable)
12. Removing registry keys prior to uninstall (if so configured)
13. Unzipping of 7z/zip archives (if so configured)
14. Performing uninstall(s)
15. Copying/Moving of remaining data directory/directories (if so configured and applicable)
16. Cleaning up old data and program directory (if so configured and applicable/viable)
17. Securing old notes.ini files and deleting them from old directories (if so configured)
18. Performing install(s)
19. Roaming of secured notes.ini files back into data directories, including adjustment of any relevant entries; also optionally includes merging of .properties and .txt files as provided in CopyTo... directories

DISCLAIMER

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

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

Although all due care has been taken in the preparation and presentation of this documentation, 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.

Limitation of liability for external links

This documentation contains links to the websites of third parties ("external links"). As the content of these websites is not controlled by panagenda, we cannot assume any liability for such external content. In all cases, the provider of information of the linked websites is liable for the content and accuracy of the information provided. At the point in time when the links were placed, no infringements of the law were recognizable to us. As soon as an infringement of the law becomes known to us, we will immediately remove the link in question.