CoreLogic®

SYMBILITY

MOBILE CLAIMS SETUP AND AUTO-UPDATE GUIDE

Author: Marc-Olivier Huynh

Revision History

Date	Description	Author(s)
July 25th, 2005	First official version	Marc-Olivier Huynh
Dec 6th, 2005	Minor updates	Marc-Olivier Huynh
Dec 7th, 2005	Added the Setup Command Line Parameters section	Marc-Olivier Huynh
Nov 29th, 2006	Added the Auto-Update section	Marc-Olivier Huynh
Oct 25th, 2007	Updated list of files that get installed Removed Pocket PC section	Marc-Olivier Huynh
Nov 21st, 2008	Refreshed hyperlink addresses Added instructions for batch deployment	Marc-Olivier Huynh
Feb 1st, 2012	New version of MFC	Marc-Olivier Huynh
June 20, 2012	Updated command line parameters section	Marc-Olivier Huynh
April 30, 2013	Updated Branding and Copyright information	Andra Postovaru
Sept 28, 2014	Updated to reflect changes included in the last major release	Marc-Olivier Huynh
Aug 19, 2015	Updated to reflect changes included in the last major release	Marc-Olivier Huynh
Jul 18, 2018	Updated URL in section 1.2	Marc-Olivier Huynh
Feb 27, 2019	Rebranding	Marc-Olivier Huynh
Jun 4th, 2021	Updated to reflect changes included in the last major release	Benoît Desmeules

Table of Contents

1. PC / Laptop / Tablet PC Setup

1.1 SETUP COMMAND LINE PARAMETERS

The available command line parameters for the Mobile Claims setup program can be found here:

http://blog.wisefaq.com/2009/01/07/all-about-installshield-command-line-switches/

Extra parameters to pass to Msiexec using /v can be found here:

http://technet.microsoft.com/en-us/library/cc759262(v=ws.10).aspx

The following example installs Mobile Claims silently in the C:\Test directory:

```
 X:\>CoreLogic_Mobile_Claims_PC.exe /s /v"/qn ALLUSERS=1 INSTALLDIR=\"C:\Test
Folder\""
```

1.2 DEPLOYING TO MANY COMPUTERS

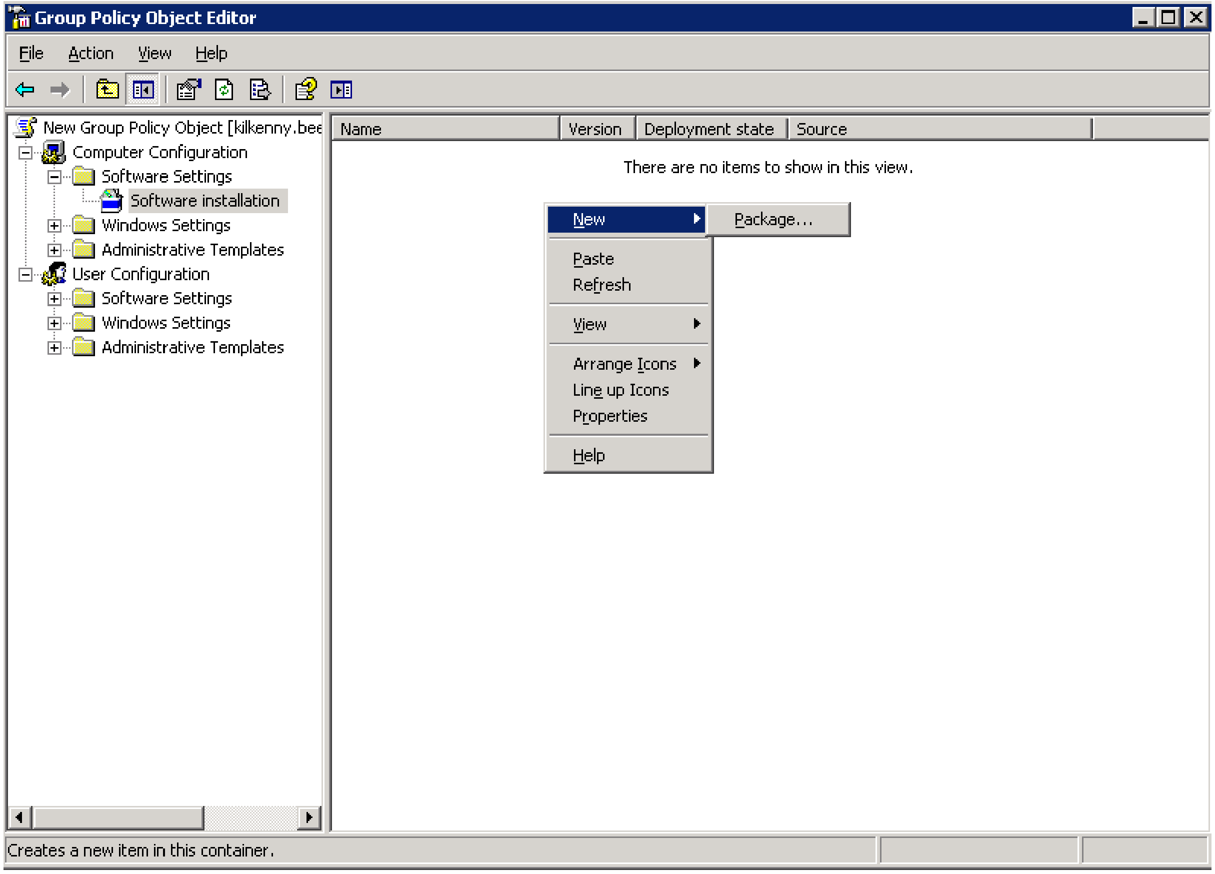
Following are the steps to deploy Mobile Claims to many computers on a domain:

1. Download the latest setup package from Claims Connect. The filename should look something like “CoreLogic_Mobile_Claims_PC_v.XX.XX.XX.exe”.
2. Open a command prompt and go to the folder containing the setup package.
3. The first step is to extract the .msi file from the package file, so type:

```
CoreLogic_Mobile_Claims_PC.exe /a
```

4. Wait until the package is done extracting, then click Next.
5. You’ll then be prompted for a destination folder. Choose a folder where you want to extract the .msi file and related files/folders from the setup package.
6. The “Mobile Claims.msi” file and 3 folders (Common, program files, System32) should now be in your destination folder.
7. Copy the files and folders on a network share to which all users have read access.
8. From the domain controller computer, open the Group Policy Object Editor for the specific domain or organizational unit. This can be done from the Active Directory Users and Computers window: right-click on your domain, sub-domain or organizational unit, choose Properties, go to the Group Policy tab, and double-click on the group policy you want to change and/or create a new one.

9. From the Group Policy Object Editor window, go to Computer Configuration / Software Settings / Software installation, right-click in the right panel, and choose New / Package.



There are no items to show in this view.

Creates a new item in this container.

10. Go to the network share where you copied .msi package and select it.

11. Choose “Assigned” and click OK.

12. Mobile Claims will be installed automatically when the computers using that Group Policy are rebooted.

13. If there are errors during the installation, they will appear in the event log of the client computers.

You can also refer to https://support.microsoft.com/en-ca/help/816102/how-to-use-group-policy-to-remotely-install-software-in-windows-server or https://www.advancedinstaller.com/user-guide/tutorial-gpo.html for more information regarding deploying .msi files to many computers.

1.3 DIRECTORIES CREATED DURING SETUP

Following is the list of directories that get created when installing Mobile Claims.

Directory	Details

\Program Files\CoreLogic Mobile Claims	This is the Mobile Claims installation directory. It can be customized during the setup. The setup program sets special rights for that folder so that normal users have read/write capabilities.

1.4 FILES INSTALLED DURING SETUP

Following is the list of files that get installed with Mobile Claims. [InstallationDir] refers to the Mobile Claims installation directory (\Program Files\CoreLogic Mobile Claims by default). [System32] refers to the Windows system directory.

Filename	Location	Details
en_US.bin	[InstallationDir]	
[xx_XX.dll] [xx_XX.aff] [xx_XX.dic] [xx_XX_README.txt]	[InstallationDir]	Files containing the resources for the language. The files included in the setup package depend on the environment. For example in the US, the files en_US.dll, en_US.aff, en_US.dic, en_US_README.txt are included.
escapi.dll	[InstallationDir]	
libeay32.dll	[InstallationDir]	
libexpatw.dll	[InstallationDir]	
libhunspell.dll	[InstallationDir]	
libpng16.dll	[InstallationDir]	
mClaim.exe	[InstallationDir]	
mClaimSetup.dll	[InstallationDir]	
mClaimUpdate.dll	[InstallationDir]	
mClaimUpdate.exe	[InstallationDir]	
mClaimUpdateCall.exe	[InstallationDir]	
ssleay32.dll	[InstallationDir]	
xmlparse.dll	[InstallationDir]	
xmltok.dll	[InstallationDir]	
zlibwapi.dll	[InstallationDir]	

cef.pak	[InstallationDir]	Chromium
chrome_elf.dll	[InstallationDir]	Chromium
ChromiumHelper.exe	[InstallationDir]	Chromium
en-US.pak	[InstallationDir]	Chromium
icudtl.dat	[InstallationDir]	Chromium
libcef.dll	[InstallationDir]	Chromium
snapshot_blob.bin	[InstallationDir]	Chromium
v8_context_snapshot.bin	[InstallationDir]	Chromium
api-ms-win-core-console-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-datetime-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-debug-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-errorhandling-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-file-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-file-l1-2-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-file-l2-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-handle-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-heap-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-interlocked-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-libraryloader-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-localization-l1-2-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-memory-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-namedpipe-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-processenvironment-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-processthreads-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-processthreads-l1-1-1.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-profile-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-rtlsupport-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-string-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-synch-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-synch-l1-2-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-sysinfo-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-core-timezone-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0

api-ms-win-core-util-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-conio-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-convert-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-environment-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-filesystem-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-heap-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-locale-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-math-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-multibyte-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-private-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-process-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-runtime-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-stdio-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-string-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-time-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
api-ms-win-crt-utility-l1-1-0.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
mfc140u.dll	[InstallationDir]	Microsoft Foundation Classes 14.0
msvcp140.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
msvcr120.dll	[InstallationDir]	Microsoft C Runtime Library 12.0
msvcr90.dll	[InstallationDir]	Microsoft C Runtime Library 9.0
Microsoft.VC90.CRT.manifest	[InstallationDir]	
ucrtbase.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
vcruntime140.dll	[InstallationDir]	Microsoft C Runtime Library 14.0
MDAC 2.8	[System32]	Microsoft Data Access Components 2.8 This is a prepackaged component that automatically installs itself if needed
MS Jet Engine 4.0 SP7	[System32]	Microsoft Jet Database Engine 4.0 This is a prepackaged component that automatically installs itself if needed
MS C Runtime Library 6.0	[System32]	Microsoft C Runtime Library 6.0

		This is a prepackaged component that automatically installs itself if needed. It extracts the file: msvcrt.dll.
MS C++ Runtime Library 6.0	[System32]	Microsoft C++ Runtime Library 6.0 This is a prepackaged component that automatically installs itself if needed. It extracts the file: msvcp60.dll.
MS Tablet PC Platform Runtime	[System32]	Microsoft Tablet PC Platform Runtime This is a prepackaged component that automatically installs itself if needed. It extracts the files: gdiplus.dll, wisptis.exe, inked.dll, inkobj.dll and tpcps.dll.

1.5 SHORTCUTS CREATED DURING SETUP

Following is the list of shortcuts that get created during the Mobile Claims setup.

Shortcut
Start Menu\Programs Menu\Mobile Claims
Desktop\Mobile Claims

1.6 REGISTRY KEYS

Following is the list of registry keys that get created during the Mobile Claims setup. At runtime, sub-keys are created and values are set.

Registry Key
HKEY_CURRENT_USER\Software\Symbility Solutions Inc.\Symbility Mobile Claims
HKEY_LOCAL_MACHINE\Software\Symbility Solutions Inc.\Symbility Mobile Claims

1.7 DIRECTORIES CREATED AT RUNTIME

Following is the list of directories that get created when Mobile Claims runs.

Directory	Details
[InstallationDir]\AutoSave	This folder is used to store temporary files.
[InstallationDir]\Temp	This folder is used to store temporary files.
[InstallationDir]\Updates	This folder is used to store temporary files.

1.8 FILES CREATED AT RUNTIME

Following is the list of files that get created when Mobile Claims runs.

Filename	Location	Details
[xx_XX.dll] [xx_XX.aff] [xx_XX.dic] [xx_XX_README.txt]	[InstallationDir]	These are the files containing the resources for each language. They get created when the user decides to download other languages from the UI in Mobile Claims.
mClaim.db	[InstallationDir]	This is the main database file. It gets created the first time a user logs in to Mobile Claims on the device.
mClaimLog.db	[InstallationDir]	The database file that stores the application logs.
mClaim.ini	[InstallationDir]	This file identifies the Mobile Claims installation.
ItemDatabase[*].db		Secondary database files that can get created when the user synchronizes Mobile Claims.
temporary files	[InstallationDir]\ AutoSave	Temporary files get created and deleted in this folder.
temporary files	[InstallationDir]\ Temp	Temporary files get created and deleted in this folder.
temporary files	[InstallationDir]\ Updates	Temporary files get created and deleted in this folder.

2. Auto-Update Procedure

2.1 INTRODUCTION

The Symbility system is comprised of two components: Claims Connect and Mobile Claims. Claims Connect is a web application; it's the communications hub and the data center. Mobile Claims is the mobile/client application that is used on the field; it synchronizes itself with Claims Connect on demand (i.e. when the user pressed the Synchronize button). Mobile Claims is tightly coupled to Claims Connect because it exchanges data with Claims Connect and needs to understand that data at all times.

Symbility has therefore decided to build within its application a very simple / dead-easy auto-update procedure. That auto-update procedure allows the Mobile Claims clients to stay up to date with the latest fixes, patches and new features.

2.2 PROCEDURE

Following are the steps that occur during a Mobile Claims auto-update.

#	Step Description
1	User X is currently working on Mobile Claims.
2	Symbility releases a new version of Claims Connect and Mobile Claims in order to deploy new features and fixes. Note that new releases always go through intense testing and quality assurance steps before being deployed to the production environment.
3	User X clicks on the Synchronize button in order to upload his changes and download the latest claims. Note that all synchronization between Mobile Claims and Claims Connect is done through web services (SOAP through HTTPS).
4	The first step of the synchronization consists in Mobile Claims sending its version number (X.X.X) to Claims Connect.
5	Claims Connect checks if there are updates available for that specific version. If there are, Claims Connect sends back a list of URLs where to download the file updates from, and sends back data telling Mobile Claims if the update is mandatory or not.
6	User X is prompted to accept of not the auto-update. If the update is mandatory, he won't be able to synchronize his data with Claims Connect until he decides to auto-update. Note that meanwhile, there's nothing preventing him to continue working (disconnected) on Mobile Claims.
7	If User X has accepted the auto-update, Mobile Claims downloads (HTTP/HTTPS) all the files required to the [InstallationDir]\Updates folder.
8	Once downloaded, Mobile Claims copies the files to the [InstallationDir] folder and overwrites/updates the appropriate files.

	Because Mobile Claims only works within its [InstallationDir] folder, NO administrator rights are required for the auto-update, and NO reboot is required. The auto-update procedure is very basic: it doesn't touch/affect the system files and the registry. The auto-update process runs as the current Windows user and is limited by the rights of that user -- it doesn't need special credentials or privileged rights.
9	Mobile Claims then restarts itself. During the restart, it checks if the local database needs to be updated. If so, it runs some database update scripts to update it.
10	User X is now back to his claim list. He can continue working and synchronize.