

|  |  |
| --- | --- |
|  | **imotion** |
|  |  |
|  | Update |
|  |  |
|  | Installation |
|  |  |
|  | Technical Description |
|  |  |
|  | Version: 2.0 |
|  | State: Released |
|  | Classification: Internal use only |
|  | Author: COS |
|  | Creation date: 2012-08-08 |
|  | Repository: $/Gorba/Main/Motion/SystemManager/Documents/TD\_AbuDhabi\_Update\_Installation.docx |
|  | Gorba AG  Sandackerstrasse  9245 Oberbüren  Switzerland |

**Table of contents**

[1 Introduction 4](#_Toc350956367)

[1.1 Purpose 4](#_Toc350956368)

[1.2 Intended Audience and Reading Suggestions 4](#_Toc350956369)

[2 Requirements 4](#_Toc350956370)

[3 Description 4](#_Toc350956371)

[4 Get the Software 4](#_Toc350956372)

[5 Software Configuration 6](#_Toc350956373)

[5.1 Update.xml 6](#_Toc350956374)

[5.1.1 <Update> 7](#_Toc350956375)

[5.1.2 <StartUpDelay> 7](#_Toc350956376)

[5.1.3 <Roots> 8](#_Toc350956377)

[5.1.4 <Root> 8](#_Toc350956378)

[5.1.5 <Copy> 8](#_Toc350956379)

[5.1.6 <Source> 9](#_Toc350956380)

[5.1.7 <Destination> 9](#_Toc350956381)

[5.1.8 <Execution> 9](#_Toc350956382)

[5.1.9 <StartUp> 9](#_Toc350956383)

[5.2 NLog.config 10](#_Toc350956384)

[6 Installation 10](#_Toc350956385)

[6.1 Without TftStartup.bat 10](#_Toc350956386)

[6.2 Using TftStartup.bat 11](#_Toc350956387)

[7 Update Example 11](#_Toc350956388)

[7.1 Update of Software and Data 11](#_Toc350956389)

[7.1.1 From USB stick 16](#_Toc350956390)

[7.1.2 From Local Directory 20](#_Toc350956391)

[7.2 Update of Update 20](#_Toc350956392)

[7.2.1 From USB stick 21](#_Toc350956393)

[7.2.2 From Local Directory 24](#_Toc350956394)

**Modification management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Modifications** | **State** |
| 0.1 | 2012-08-08 | COS | SW | Initial version | draft |
| 0.2 | 2012-09-12 | RAN | SW | Changed the version number and path of release of Update.exe. Changed StartupDelay default value to 120. Added information about update with TFTStartup.bat. | draft |
| 1.1 | 2013-03-13 | COS | SW | Added the chapter 3 to give a brief description of the software.  Add notes according to the email of SEM of 17 Dec. 2012 | draft |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  | 2013-03-13 | RAN | SW | Minor changes to chapter 3, 4 |
| 1.1 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  | 2012-09-12 | RAN | SW | First release version |
| 1.0 |  |  |  |  |
| 2.0 | 2013-03-13 | RAN | SW |  |
|  |  |  |  |  |

**References**

|  |  |
| --- | --- |
| **ID** | **Path** |
|  |  |
|  |  |

# Introduction

## Purpose

The goal of this document is to describe in detail everything is required to install, configure and run Update.exe starting from its version number **1.0.1237.2457**.

## Intended Audience and Reading Suggestions

This document is addressed to product managers or people that are familiar with this product and are able to install, operate and maintain it.

# Requirements

Before starting to install Update.exe, make sure to have the following requirements:

1. Windows XP, Windows XP Embedded, Windows Vista 32/64 bit, Windows 7 32/64 bit
2. .NET Framework 2.0 installed
3. At least 10 MB of space in the hard disk for Update.exe

If you don’t have the .NET Framework 2.0, you can download it from:   
<http://www.microsoft.com/download/en/details.aspx?id=19>

In the rest of this document the target PC is a TopBox with O.S. Windows XP Embedded in English language.

# Description

This software, as its name says, has the main goal to update the content of specific directories with files coming from other specific directories. The directories from which are coming the *new* files represent the “source” of the update phase and the directories in which will be put the *new* files are the “destination”: basically, Update.exe is a clever *directories replacer*. As additional feature, it can also execute a second application or script after the directory update process. In this way, if configured correctly, Update.exe can also be thought of as a bootstrapper of other additional software.

The source directories can be local or can be on a USB pen drive: Update.exe doesn’t care about the source disk type (a hard disk or a removable one); it searches for specific directories and once found, starts its process. Usually the *new* files are stored in a USB pen drive and the destinations are directories in the hard disk of the PC, but nobody avoids configuring Update.exe with different combinations. Also, in the AbuDhabi system, Update.exe is configured to be the first running application due to its bootstrapper capability.

During an update process several problems can occur: for example the source can be suddenly removed, the USB pen drive used as source can “disappear” from the list of the disks recognized by the O.S. and so on… In order to avoid loss of information, Update.exe does a backup of the destination directories before really starting. The backup is not done coping and pasting the directories somewhere else but rather renaming them. It this way it doesn’t need additional space in the hard disk for the backup because the original directories are only renamed. In case of failure, Update.exe reestablishes the original name. If you kill Update.exe during its executing, the backed up directories are left with their new name and the only way to restore the correct ones is doing it manually.

Upon completion of the update process, it launches the configured application, usually SystemManager.exe.

# Get the Software

The latest delivered version of Update.exe is downloadable from the following path:

[\\Softwareserver\Release\SW\02\_imotion\02\_TFT\12\_SystemManager](file:///\\Softwareserver\Release\SW\02_imotion\02_TFT\12_SystemManager)

Update.exe is part of the SystemManager.exe package. They cooperate in the following way:

* Update.exe updates (eventually) the system with new files after every system’s boot
* Update.exe runs SystemManager.exe
* SystemManager.exe runs other applications (usually Protran, MediServer and InfoMedia) and acts as their watchdog.

Inside that directory there are all the binary files and all the configuration files required to install Update.exe properly.

Make sure to have at least the following set of files:

|  |  |
| --- | --- |
| **File** | **Description** |
| Update.exe | This file launches in execution the software as a console application. All the logs produced during the Update’s activity can be viewed here (if enabled in NLog.config). |
| Gorba.Common.Configuration.dll | Library containing the iMotion’s *engine* required to load configuration file. |
| Gorba.Motion.SystemManager.Update.Core.dll | Library containing all the logic that effectively realizes the update process. |
| Gorba.Motion.SystemManager.Core.XmlSerializers.dll | Library needed by Update.exe to speed up it Library needed by |
| Gorba.Motion.SystemManager.Update.Core.XmlSerializers.dll | Update.exe to speed up its boot s boot |
| NLog.dll | Library for the application’s log activities. |
| NLog.config | Configurations file for the library NLog. |
| Update.xml | Configurations file for the executable Update.exe |

Table 4‑1

Also, make sure that the Update.exe version is bigger or equal than **1.0.1237.2457**:

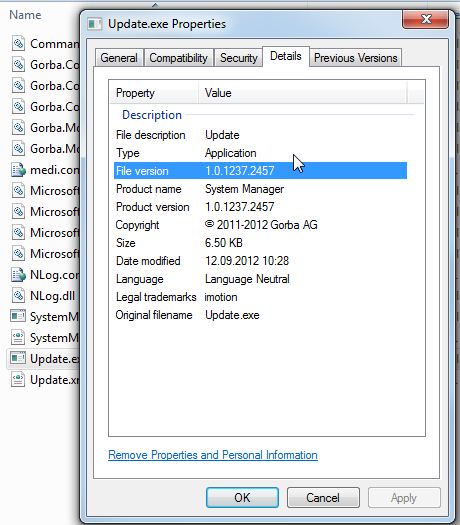


Figure 4‑1 Update.exe version

Once you checked all the things mentioned above, you can proceed with the Update.exe configuration and installation.

# Software Configuration

Update.exe needs two configuration files:

* Update.xml
* NLog.config

The first file contains all the information required by the software in order to perform the update process: backup operations, deletions, copies and so on.

The second file contains all the information about the software’s logging activities.

## Update.xml

A complete example of Update.xml is shown here below:

<?xml version="1.0" encoding="utf-8" ?>

<Update>

  <StartUpDelay>120</StartUpDelay>

  <Roots>

    <Root DeleteAfterUpdate="false">F:\UpdateData\</Root>

    <Root DeleteAfterUpdate="false">E:\UpdateData\</Root>

    <Root DeleteAfterUpdate="true">D:\UpdateData\</Root>

  </Roots>

  <Copy>

    <Source>Progs</Source>

    <Destination>D:\Progs</Destination>

  </Copy>

  <Copy>

    <Source>InfoMedia</Source>

    <Destination>D:\InfoMedia</Destination>

  </Copy>

  <Execution>Scripts\Update.bat</Execution>

  <StartUp>D:\Progs\SystemManager\SystemManager.exe</StartUp>

</Update>

Figure 5‑1 Update.XML complete example

A formal description of all the tags shown above is in the following paragraphs.

### <Update>

This is the document’s root.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| No | Yes | Just the container of all the other tags. | No value. Only its allowed sub-tags. |

Table 5‑1 Update Tag

Sub-tags allowed:

* <StartUpDelay> (only one is accepted)
* <Roots> (only one is accepted)
* <Copy> (multiple tag of this type are allowed)
* <Execution> (only one is accepted)
* <StartUp> (only one is accepted)

### <StartUpDelay>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| No | Yes | It represents the maximum amount of seconds in which Update.exe tries to detect the valid source that will be used for the update process. Only if the source is detected before that limit of time, Update.exe starts immediately its tasks, otherwise it waits until the end. | Integers from 1 to 65536. The default value is 30. |

Table 5‑2 StartUpDelay Tag

Sub-tags allowed: none.

### <Roots>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| No | Yes | This XML tag contains the set of all the **potential** sources for the update process. It’s extremely important the order in which the <Root> are inserted: will be considered valid only **the first** source detected in a top-down order. | No value. Only its allowed sub-tags. |

Table 5‑3 Roots Tag

Sub-tags allowed:

* <Root>

Remarks:

If for example Update.exe finds that the source “F:\UpdateData\” has directory(ies) and/or batch file(s) to be updated, it will select “F:\UpdateData\” and the others remaining “E:\UpdateData\” and “D:\UpdateData\” will not be taken in consideration.

### <Root>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Meaning** | **Values admitted (default value)** |
| Yes | Yes | Container of the information about where Update.exe has to search for the directories to be updated. | The absolute path of a directory to be considered as source. The default value is an empty string. |

Table 5‑4 Root Tag

Sub-tags allowed: none.

Attributes allowed:

* DeleteAfterUpdate

Attention:

* it’s strongly suggested to set to “true” the attribute DeleteAfterUpdate **if the root points to a directory stored in the hard disk of the PC**. After an update, if a local root is not deleted, the update process will take place again with the new restart of the PC.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Mandatory** | **Meaning** | **Values admitted (default value)** |
| DeleteAfterUpdate | Yes | Tells if the source has to be deleted after the update process. Attention: it’s strongly suggested to set to “false” this attribute when doing the update via USB stick. | The strings “true” or “false”.  Default value “false”. |

Table 5‑5 DeleteAfterUpdate Attribute

### <Copy>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| No | Yes | Contains all the information required to move files from the source to a specified destination. | No value. Only its allowed sub-tags. |

Table 5‑6 Copy Tag

Sub-tags allowed:

* <Source>
* <Destination>

### <Source>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| No | Yes | Contains the name of the directory in the source that will be copied to a specific destination during the update process. | A string. The default value is an empty string. |

Table 5‑7 Source Tag

Sub-tags allowed: none

### <Destination>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| No | Yes | Contains the name of the directory in which will be copied a specific directory in the source. If the destination directory doesn’t exist, it will be created. | A string. The default value is an empty string. |

Table 5‑8 Destination Tag

Sub-tags allowed: none

### <Execution>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| Yes | Yes | Contains the path (in the source) of the batch file to be executed after the update process when is needed to update the software Update.exe itself. | A string. The default value is an empty string. |

Table 5‑9 Execution Tag

Sub-tags allowed: none

Attributes allowed:

* args

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Mandatory** | **Meaning** | **Values admitted (default value)** |
| args | Yes | The arguments line to be used to execute the batch file. | A string. The default value is an empty string. |

Table 5‑10 args Attribute

### <StartUp>

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag’s attributes** | **Mandatory** | **Tag’s Meaning** | **Values admitted (default value)** |
| Yes | Yes | Contains the absolute path of the application to be executed after the update process: this application is the one tasked to run Protran, InfoMedia and MediServer and to act as the system’s watchdog. | A string. The default value is an empty string. |

Table 5‑11 StartUp Tag

Sub-tags allowed: none

Attributes allowed:

* args

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Mandatory** | **Meaning** | **Values admitted (default value)** |
| args | Yes | The arguments line to be used on executing the application tasked to run Protran, InfoMedia and MediServer and to act as the system’s watchdog. | A string. The default value is an empty string. |

Table 5‑12 args Attribute

## NLog.config

A complete example of NLog.config is shown as follows:

Figure 5‑2 NLog.config complete example

<?xml version="1.0" encoding="utf-8" ?>

<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"

      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <targets async="true">

    <target xsi:type="File" name="file" fileName="update.log"

            layout="${longdate} ${uppercase:${level}} ${logger} [${threadid}:${threadname}] ${message}${onexception: ${newline}${exception:format=tostring}}"

            archiveAboveSize="5120000"

            archiveNumbering="Sequence"

            concurrentWrites="true"

            keepFileOpen="false" />

    <target xsi:type="ColoredConsole" name="console"

            layout="${time} &lt;${logger:shortName=true}&gt; ${message}${onexception: ${newline}${exception:format=tostring}}" />

  </targets>

  <rules>

    <logger name="\*" minlevel="Info" writeTo="file" />

    <logger name="\*" minlevel="Info" writeTo="console" />

  </rules>

</nlog>

In the example above, the logger is configured to write both on the console and in the file “update.log” (in append mode). In order to change where Update.exe has to log (on the console and/or on a file) and the verbosity, it’s enough to change these two properties here: “minLevel” and “writeTo”.

The available levels of verbosity (ordered from the most verbose to the smallest one) are:

* Off
* Trace
* Debug
* Info
* Warn
* Error
* Fatal

# Installation

There are two possible ways to install Update application on the Gorba Topbox.

## Without TftStartup.bat

For the rest of this document will be assumed that the target computer is a Gorba’s TopBox with Windows XP Embedded in English language, the user’s account is “InfoMedia” and the installation directory is **D:\Update**. Whenever is specified *[Update]*, the user has to replace it with the proper installation directory (in this chapter is D:\Update).

To install Update are required the following steps:

1. Get the software as specified in the chapter 3
2. Copy all required file in *[Update]*
3. Create a shortcut of *[Update]*\Update.exe
4. Move the shortcut created in the previous step at the path:

C:\Documents and Settings\InfoMedia\Start Menu\Programs\Startup

1. Commit the changes did on the C:\ drive opening a prompt shell, editing the command “commit” and pressing enter.

As result, you have the whole Update software ready to run, and also, it will be also executed automatically after each system startup.

## Using TftStartup.bat

To update the Gorba Topbox from using TFTUpdate to using Update.exe, refer to the document TD\_Topbox Update using TFTStartup.pdf. This process of update is performed only once to switch the Gorba Topbox from TFTUpdate to Gorba’s applications including Update.exe.

# Update Example

In this chapter will be shown examples of usage of Update.exe in order to:

* Update the software and data using a USB stick
* Update the software and data using a local directory already present in the TopBox
* Update the Update software using a USB stick
* Update the Update software using a local directory already present in the TopBox

All the examples have screenshots taken directly from a TopBox in which was installed Update.exe (following what was explained in the chapter 6).

## Update of Software and Data

Before the update process, the situation of the drive **D:\** in the TopBox is the following:

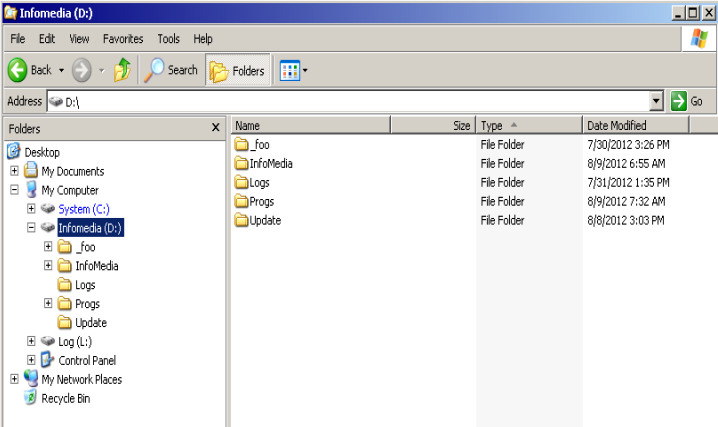


Figure 7‑1 Initial situation of the D:\ drive in the TopBox

As you can see, the TopBox has already the directory **D:\InfoMedia** in which are installed all the data (images, videos, etc…) useful for the Gorba’s media player, the directory **D:\Progs** in which are installed all the (old) software required for the AbuDhabi project and the directory **D:\Update** in which is installed Update.exe.

In particular, the directory **D:\Progs** has the following sub-directories:

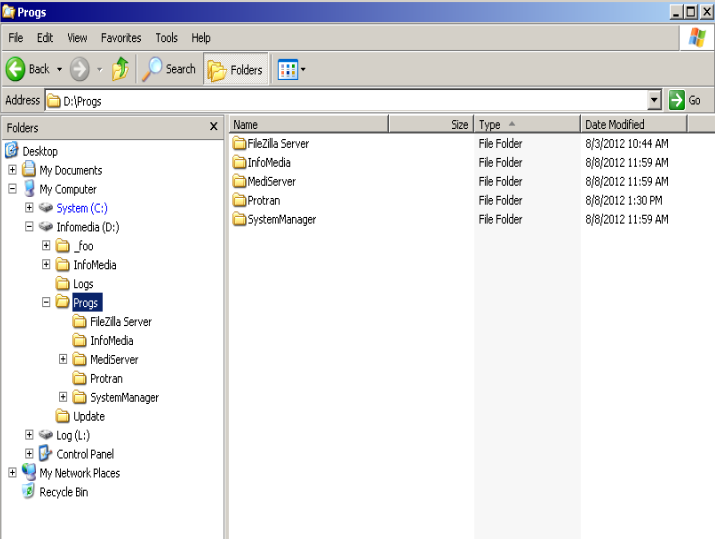


Figure 7‑2 Initial situation of the D:\Progs drive in the TopBox

The set of required software for the AbuDhabi project is composed by the FTP server “FileZilla Server.exe”, the media player “InfoMedia.exe”, the messages dispatcher “MediServer.exe”, the IBIS/ISI protocol translator “Protran.exe” and the system’s watchdog “SystemManager.exe”.

The versions of the currently installed software are:

* InfoMedia 1.0.104.21
* MediServer 1.0.1231.2073
* Protran 1.4.1231.2099
* SystemManager 1.0.1231.2108

as shown in the following screenshot:

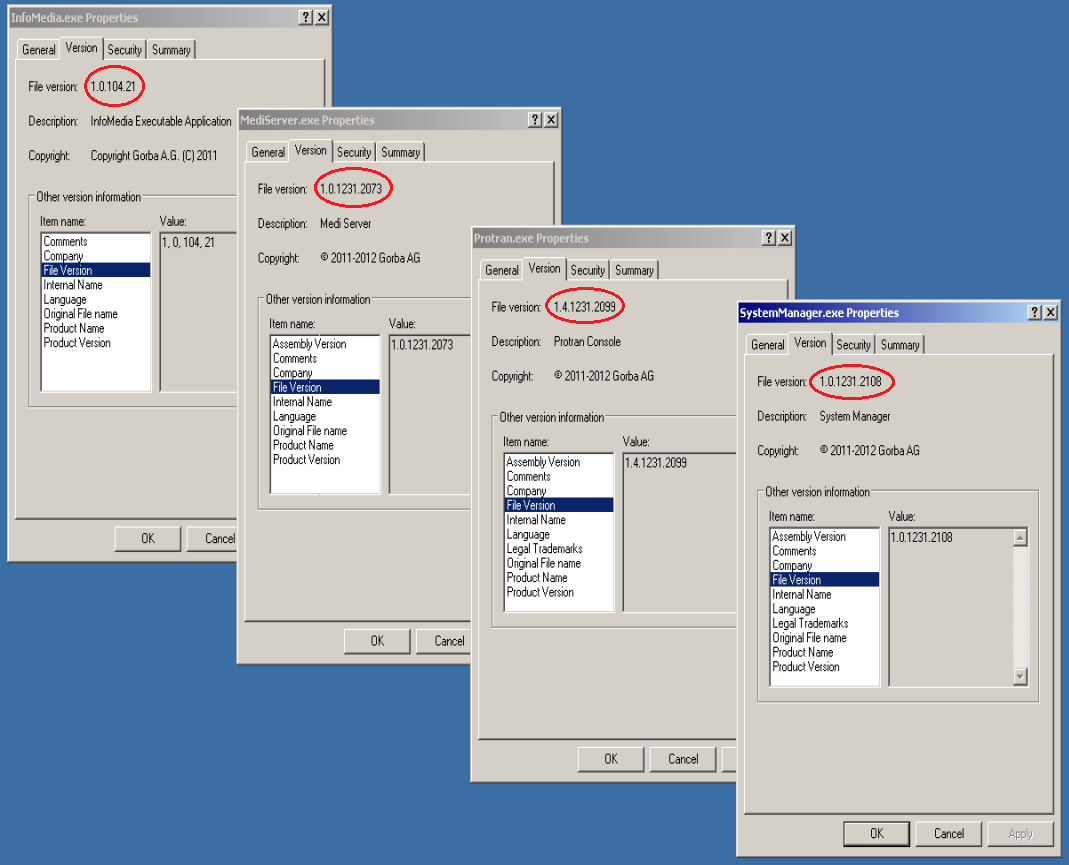


Figure 7‑3 Initial versions of the software in D:\Progs

The **D:\InfoMedia** directory currently contains the layout used for the BLT project. This is shown in the following screenshot:

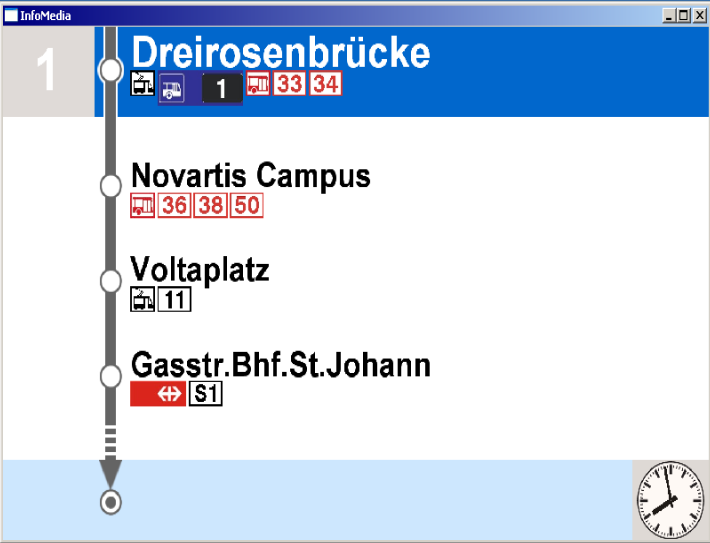


Figure 7‑4 Initial layout for InfoMedia.exe in D:\InfoMedia

The goal of Update.exe is to update the directory **D:\Progs** with newer software and the directory **D:\InfoMedia** with the layout used for the AbuDhabi project. Update.exe is configured as shown here below:

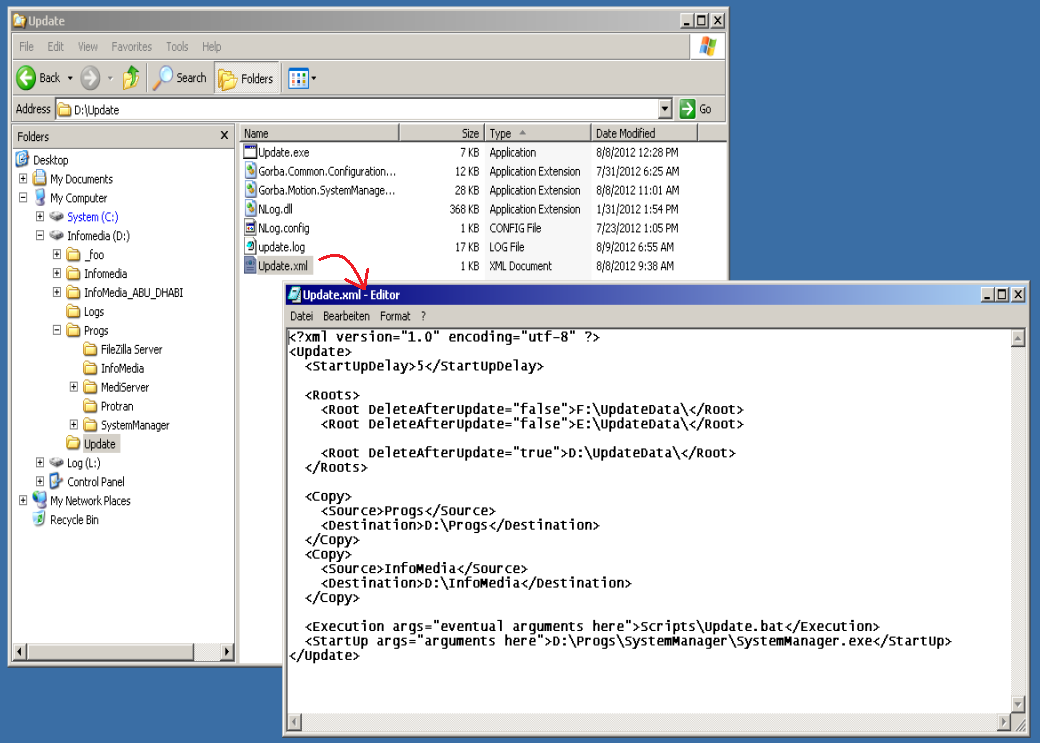


Figure 7‑5 Configuration of Update.exe

The goal of Update.exe can be achieved in the two strategy detailed in the next two paragraphs.

### From USB stick

In order to update the software and the data of the TopBox, is mandatory to prepare a USB stick as shown in the following screenshot:

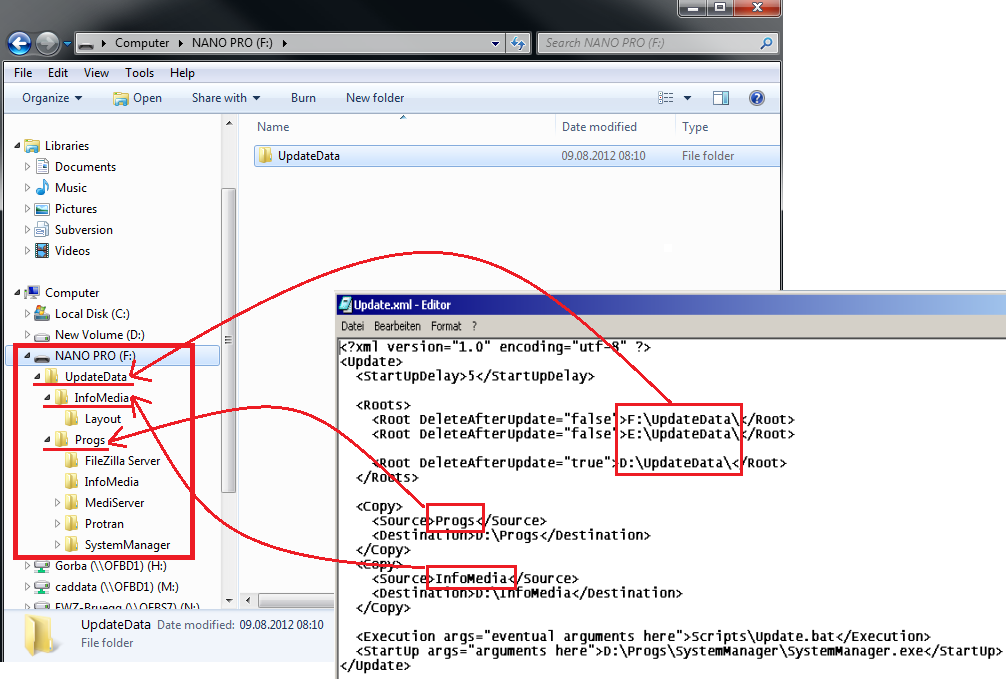


Figure 7‑6 Preparation of the USB stick to update **D:\Progs** and **D:\InfoMedia**

When the USB stick will be plugged in the TopBox, it will be detected by the O.S. as a drive and will receive a drive letter. Typically E:\ (the next available letter after “D”) but this is not guaranteed. For this reason is advisable to specify in Update.xml more than one root source (but always paying attention to the top-down order used by Update.exe).

The directory **F:\UpdateData\InfoMedia** contains the layout for the AbuDhabi project and it will be copied (with all its subdirectories) in **D:\InfoMedia**, according to the Update.exe configuration file.

The directory **F:\UpdateData\Progs** contains all the new software and it will be copied (with all its subdirectories) in **D.\Progs**, according to the Update.exe configuration file.

Once the USB stick is prepared, it’s the time to plug it into the TopBox.

The procedure to follow is:

* Make sure that the TopBox is shut down
* Plug the USB stick in one of the USB slot of the TopBox
* Start the TopBox

During the update process, Update.exe produces some logs (according to its NLog.config file) as the following ones:



Figure 7‑7 Logs of Update.exe

As you can see in the figure above, Update.exe has detected the USB stick mapped to drive E:\ and has copied the direcotories **E:\UpdateData\Progs** to **D:\Progs** and **E:\UpdateData\InfoMedia** to D:\InfoMedia. In addition, after the update phase it has also executed the application **D:\Progs\SystemManager\SystemManager.exe** and finally has exited.

Now, what is running in the TopBox is shown here below:

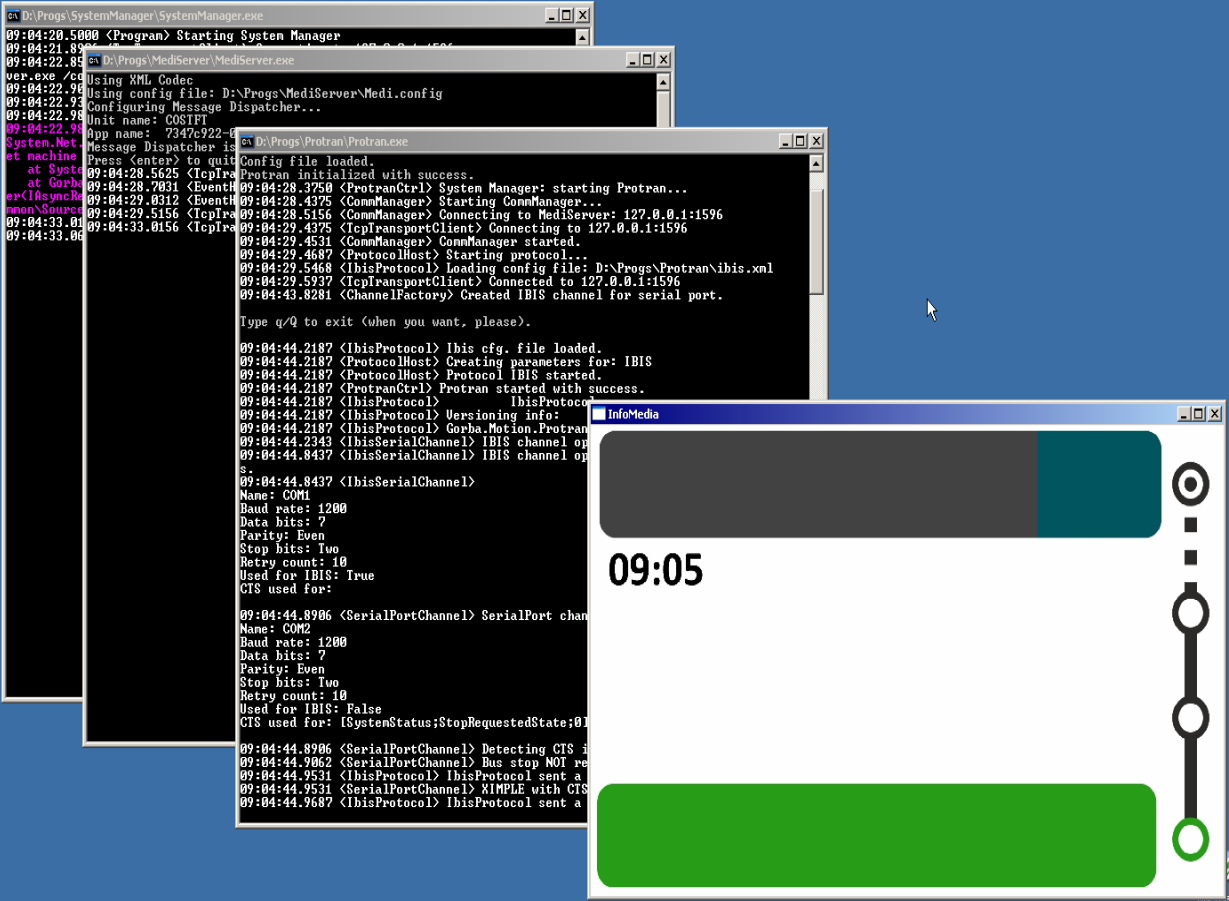


Figure 7‑8 Applications running after the update process

You can see immediately that InfoMedia.exe is now loading the typical layout of the AbuDhabi project and not anymore the one for the BLT project. Once you see that all the applications are running properly, you can assume that the update process is finished with success.

Now the new software have the versions:

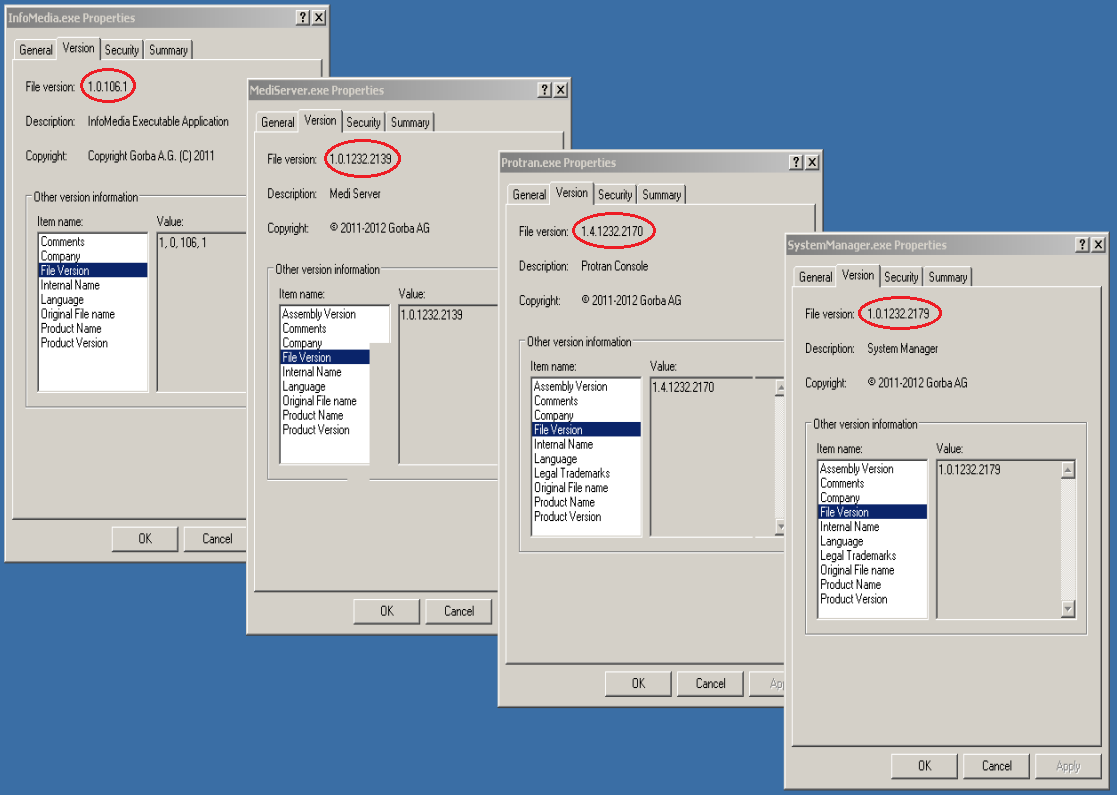


Figure 7‑9 The new updated software

The versions of the new installed software are:

* InfoMedia 1.0.106.1
* MediServer 1.0.1232.2139
* Protran 1.4.1232.2170
* SystemManager 1.0.1232.2179

### From Local Directory

The procedure to update the TopBox software and data from the local directory is equal to one explained in the previous paragraph except that the new software and the new data must be placed not in a USB stick but instead in the directory **D:\UpdateData** of the TopBox. After the update process, the source **D:\UpdateData** will be cleared of its content, accordingly with the configuration file Update.xml.

## Update of Update

Before the update process, the Update.exe situation in the TopBox is the following:

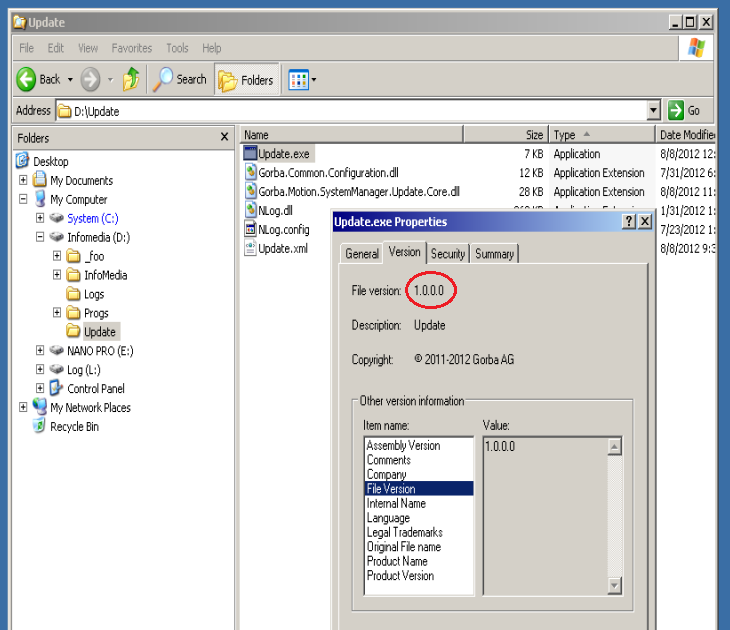


Figure 7‑10 Initial situation of the Update.exe in the TopBox

As you can see, the version of Update.exe is 1.0.0.0. It will be updated with a newer (and official) one.

Update.exe is configured with the same Update.xml file used in the previous example.

The goal of Update.exe can be achieved in the two strategy detailed in the next two paragraphs.

### From USB stick

In order to update the Update software installed in the TopBox, is mandatory to prepare a USB stick as shown in the following screenshot:

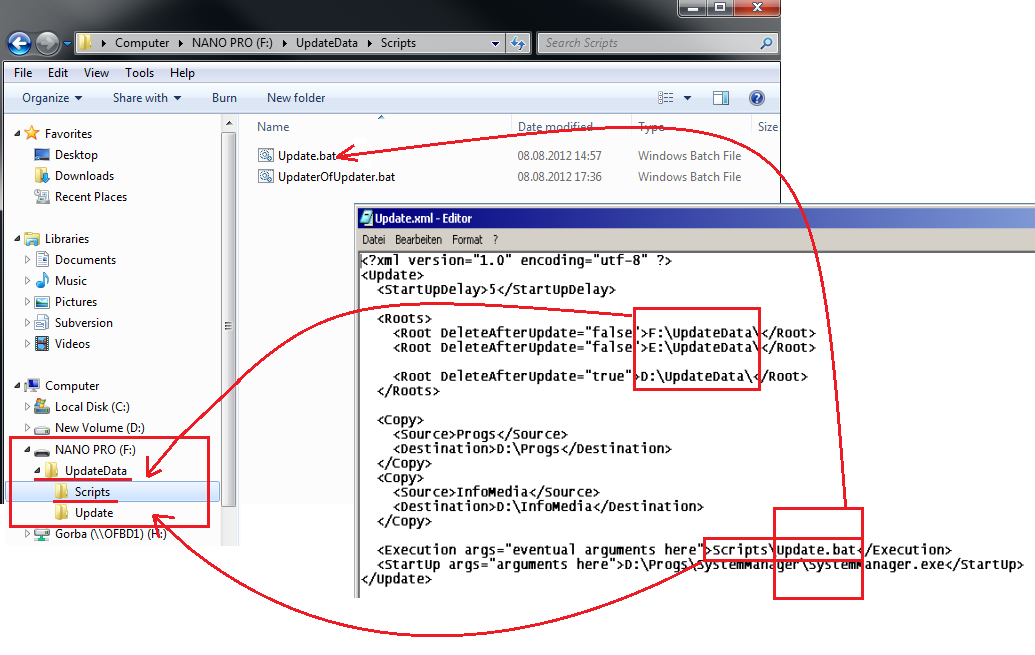


Figure 7‑11 USB stick with the required files to update the Update software in the TopBox

As you can see, accordingly to the configuration file Update.xml, in the directory **F:\UpdateData\Scripts** is stored the file **Update.bat**. This file will be executed by Update.exe, then Update.exe will close itself and the batch **Update.bat** will call the second batch **UpdaterOfUpdater.bat**. Actually this is the batch file that does the real update process but the combination of both of them is mandatory.

The new binary files of the Update software are stored in the directory **F:\UpdateData\Update** accordingly on how is coded **UpdaterOfUpdater.bat**.

The procedure to follow is:

* Make sure that the TopBox is shut down
* Plug the USB stick in one of the USB slot of the TopBox
* Start the TopBox

During the update process, Update.exe produces some logs (according to its NLog.config file) as the following ones:

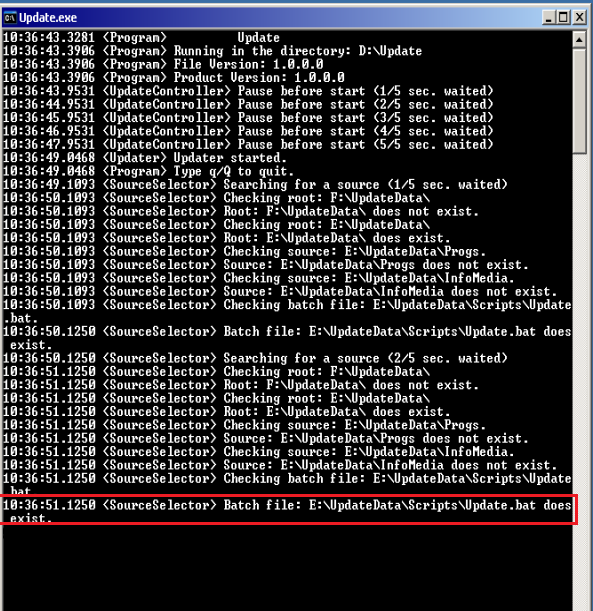


Figure 7‑12 Logs of Update.exe to update itself

As you can see in the figure above, Update.exe has detected the USB stick mapped to drive E:\ and has executed the batch file **E:\UpdateData\Scripts\Update.bat**. Finally, after the update phase the application **D:\Progs\SystemManager\SystemManager.exe** is executed. The TopBox still has the same software and data that it had before because the update process didn’t affect that parts. The only thing that is changed is the Update software itself and this is shown in the following screenshot:

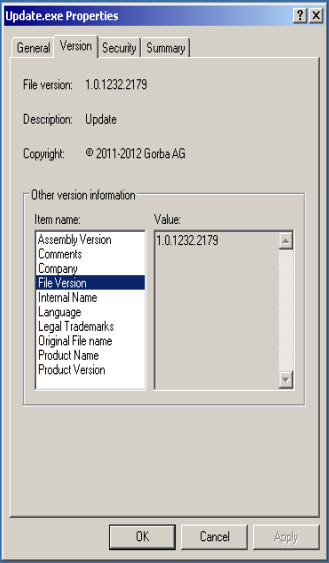


Figure 7‑13 New Update.exe installed in the TopBox

As you can see, the version of Update.exe is now 1.0.1232.2179.

The script **Update.bat** is always called by Update.exe (if opportunely configured), with or without a real update process to be done. If there’s really new data to be updated or even if there’s nothing at all, this doesn’t make any influence to the “update of update” process because it starts at the end of all.

### From Local Directory

The procedure to update the update software from the local directory is equal to one explained in the previous paragraph except that the new software and the batch files must be placed not in a USB stick but instead in the directory **D:\UpdateData** of the TopBox.

After the update process, the file **D:\UpdateData\Scripts\UpdaterOfUpdater.bat** will be deleted automatically in order to not repeat again the update process each time the system starts.