APPLICATION NOTE

One Cambridge Center, Cambridge, MA 02142 • Tel: 866-833-4069 • www.thingmagic.com 9/11

Managing the Firmware Upgrade Process with ThingMagic Enterprise Readers

By Satyan Shah

Introduction

Managing and updating firmware on large populations of networked readers can be difficult and time consuming. As RFID deployments evolve and become more complex, additional functionality like the ability to read sensors, detect counterfeit tags, or check tamper alarms, will likely be provided via firmware upgrades.

To reduce this effort and centralize the firmware upgrade process, ThingMagic's enterprise readers contain a software feature, known as 'Auto-Upgrade', This application note describes how one can manage the upgrade process using ThingMagic UHF RFID technology.

Note: *Reader and Server configurations are provided as examples in this document. End users will likely need to modify configuration options to work in their environment.*

System Components

ThingMagic's Auto-Upgrade functionality was designed to provide the most flexibility to system administrators and use existing infrastructure where available.

To take full advantage of Auto-Upgrade, a few important systems should be online and configured correctly. Each of these systems is explained in more detail below.

1. **ThingMagic Enterprise Readers**: The ThingMagic Mercury5 (M5), Astra®, and latest generation Mercury6 (M6) all support Auto-Upgrade. These proven RFID platforms offer industry leading interfaces, specifications, and performance.

2. **DHCP Server**: Most large-scale RFID reader deployments use a DHCP server to manage network addresses. Auto-upgrade builds on top of this expected infrastructure by delivering Options 66 and 67 from the DHCP server when clients request an address. These two options are

overridden and delivered by the DHCP server, and ThingMagic Enterprise Readers use this information to locate the Auto-Upgrade configuration package.

3. **TFTP Server or Static Media Server**: To minimize network bandwidth, a TFTP server or static media server can be used to deliver an Auto-Upgrade configuration package to the reader. This light-weight file will contain references to the location of the latest ThingMagic firmware and the minimum version of software that should trigger an upgrade.

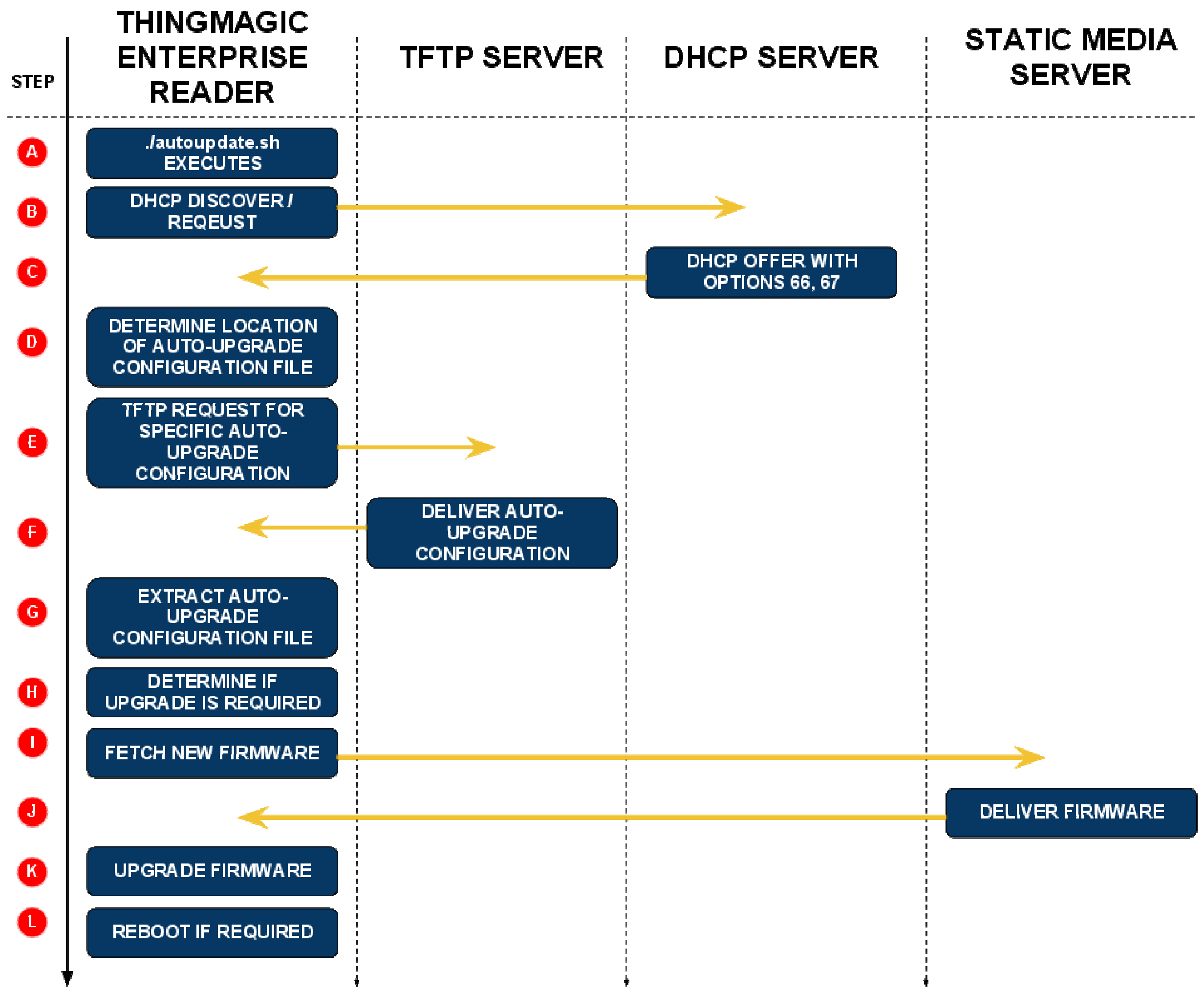
> ***NOTE****: ThingMagic believes network bandwidth is a limited resource that should be conserved whenever possible. A large population of readers, all simultaneously attempting to download a 7.5MB firmware update, could easily consume a disproportionate amount of available bandwidth on a user's network.*
>
> *To minimize network traffic, users should always make use of the TFTP server or alternative methods (described later) to deliver the intermediate Auto-Upgrade configuration package. The configuration package allows the reader to quickly determine if an upgrade is required, without fetching the entire firmware update package.*

4. **Static Media Server:** The static media server is responsible for delivering a complete build of firmware when requested. ThingMagic readers support both HTTP and FTP protocols.

Auto-Upgrade

ThingMagic's Auto-Upgrade capability makes deploying new firmware as easy and efficient possible. Each step of the upgrade process is detailed and discussed in this section.

The diagram below illustrates the sequence that takes place



STEP
THINGMAGIC ENTERPRISE READER
TFTP SERVER
DHCP SERVER
STATIC MEDIA SERVER
A
./autoupdate.sh EXECUTES
B
DHCP DISCOVER / REQEUST
C
DHCP OFFER WITH OPTIONS 66, 67
D
DETERMINE LOCATION OF AUTO-UPGRADE CONFIGURATION FILE
E
TFTP REQUEST FOR SPECIFIC AUTO-UPGRADE CONFIGURATION
F
DELIVER AUTO-UPGRADE CONFIGURATION
G
EXTRACT AUTO-UPGRADE CONFIGURATION FILE
H
DETERMINE IF UPGRADE IS REQUIRED
I
FETCH NEW FIRMWARE
J
DELIVER FIRMWARE
K
UPGRADE FIRMWARE
L
REBOOT IF REQUIRED

autoupdate.sh Executes (Step A)

The Auto-Upgrade process is controlled on the reader via a unix shell script named autoupdate.sh. This script executes at the users option. Typically, autoupdate.sh is executed on a periodic basis via cron.

Cron is a UNIX utility that allows tasks to be automatically run in the background at regular intervals by the cron daemon. These tasks are often termed as cron jobs in UNIX. Crontab (CRON TABle) is a file which contains the schedule of cron entries to be run and at specified times [Reference].

How to setup autoupdate.sh to execute on a periodic basis.

1. The first step is to open a terminal session to a ThingMagic Reader. This can be accomplished via the serial console, telnet or SSH.
2. Next, navigate to the etc directory by typing "cd /etc/" at the command prompt
3. Use vi to edit the file crontab and set the script located at /bin/autoupdate.sh to execute as desired.
4. There is a command line argument, referred to as "jitter" available to the user. When autoupdate.sh is executed with the jitter parameter specified, the reader will select a time between 0 and j and then execute the script. This parameter, designated by "-j" at the command line followed by an integer, is added to reduce network congestion.

***NOTE**: The default contents of crontab contain an entry to support periodic checks for firmware updates. This entry, however, is commented out.*

```
# Update firmware every day at 4am (600s jitter)
#* 4 * * * root /bin/autoupdate.sh -j 600
```

Simply remove the "#" character at the beginning of the second line, save the file, and the task will now execute as described.

DHCP Discover (Step B)

The reader, upon booting, broadcasts a DHCP discover packet.

DHCP Offer (Step C)

The DHCP server must respond to the reader with a DHCP Offer. The offer should contain Options 66 and Options 67.

How to setup a DHCP server to specify Options 66 and 67.

1. Select a DHCP server. For the purposes for this example the assumed configuration is Ubuntu 10 + dhcp3.
2. Install this software (you may need extra permissions to perform this step)
3. Locate your DHCP configuration file. By default this file should be stored at /etc/dhcp3/dhcpd.conf

4. Locate the options definition section of this configuration file. Declare the following two options:

```
option tftp-server code 66 = string;

option tftp-file code 67  = string;
```

5. Location the subnet declaration section of this configuration file. Declare your subnet as follows (please note it is likely your network configuration will NOT be the same as what's shown below)

```
subnet 10.0.0.0 netmask 255.255.255.0 {

range 10.0.0.120 10.0.0.140

option tftp-server = "10.0.0.111"

option tfpt-file =  "autoUpdateLite.tmfw"

}
```

> ***NOTE**: The subnet declaration above indicates that the DHCP server will offer addresses in the range of 10.0.0.120 to 10.0.0.140.*
>
> *Any client that receives a DHCP offer will also have access to the tftp-server and tftp-file options specified.*

6. Restart your DHCP server by typing sudo /etc/init.d/dhcp3 restart.

Auto-Upgrade configuration package and TFTP File Request (Steps D & E)

At this point the ThingMagic Enterprise Reader will receive a server address ("next-server", option 66) and file name ("boot-file", option 67) from the DHCP server upon initialization.

Once, the autoupdate.sh script executes, either manually or by cron, the reader will locate the TFTP server and request the file.

TFTP File Transfer (Step F)

The TFTP server must respond to the reader by delivering the requested file.

How to setup a TFTP server to specify Options 66 and 67.

1. Select a TFTP server. For the purposes for this example the assumed configuration is Ubuntu 10 + tftpd-hpa.

2. Install this software (you may need extra permissions to perform this step)
3. Locate and open the tftp-hpa configuration file located at /etc/default/tftpd-hpa
4. Configure the following 2 options:

```
TFTP_DIRECTORY=”/var/lib/tftpboot”

TFTP_ADDRESS=”10.0.0.111:69”
```

5. Move the Auto-Upgrade configuration package, “autoUpdateLite.tmfw”, to the location specified by TFTP_DIRECTORY.
6. Restart your tftp server by typing sudo /etc/init.d/tftpd-hpa restart

Alternative Method for delivering the Auto-Update configuration package (Steps C-F)

Users are not required to use a TFTP server to deliver the Auto-Update configuration package (this is the file referred to as “autoUpdateLite.tmfw” in examples). Two additional options to deliver this package are available.

1. **Using more DHCP parameters:** If users prefer not to use a TFTP server, the next-server and boot-file parameters of the DHCP server may be used.
2. **Using the boot_config parameter:** Users may also explicitly specify the server and file which hosts the Auto-Update configuration package. Users can specify the location by assigning any URL to the parameter “boot_config” in the tm.conf file on the ThingMagic Reader. It should be noted that this method would require an update to each reader’s tm.conf file should the server or file name change in the future.

Determining if an upgrade is required (Steps G and H)

Once the reader can download and extract the autoupdate.sh package, it will attempt to locate a ThingMagic configuration file named tm.conf. This file is expected to contain two important parameters:

1. **minimum_version** will be compared to the existing version of software, defined on the ThingMagic Enterprise Reader at /etc/tm/tm_version. Should the minimum_version be greater than the existing version, the upgrade process will take place.
2. **boot_firmware** will be used in the event a firmware upgrade is required. This URL is specified as a path to the full ThingMagic firmware.

If it is determined a firmware upgrade is required, subsequent steps will follow the boot_firmware URL to retrieve the firmware and install it.

In the event an update is not required, the reader will delete the autoUpdateLite.tmfw package and resume normal operation until the next time autoupdate.sh is called.

Fetching Firmware (Step I)

After the reader has determined an upgrade is required it will naturally need to download the firmware. The reader performs a HTTP request or FTP request, based on the URI specified in the boot_firmware field, to begin the file transfer.

Upgrading Firmware (Steps J and K)

ThingMagic Readers typically upgrade firmware in the background. If firmware can be updated without rebooting the device, the reader will handle this gracefully while continuing to inventory RFID tags.

Tools

ThingMagic's has created a few tools to aid in the Auto-Upgrade process. These tools are described below and available by contacting support@thingmagic.com

make_autoupdate.sh

The shell script make_autoupdate.sh allows users to automatically create an Auto-Update configuration package. The script is controlled via command line parameters and must be executed 'against' an existing ThingMagic firmware file.

Usage

To use this utility, simply type the following in your UNIX terminal:

```
./make_autoupdate.sh –m 4.7.1 tm.conf
```

The –m option will set the minimum_version field in the tm.conf file to the user specified value before creating the autoUpdateLite.tmfw package.

In the event the user wants to extract a firmware version from an existing .tmfw, the following can also be used:

./make_autoupdate.sh –m 4.7.1 –e myCurrentTMFirmware.tmfw tm.conf

The –e option can be a path to a local file or URL. The script will retrieve this file and extract the firmware version from it. The minimum_version field in the tm.conf file will be set to this value.

Summary

Upgrading firmware across large-scale reader deployments can be managed efficiently and intelligently using the Auto-Upgrade feature of ThingMagic's Enterprise Readers. Automating the firmware upgrade process can save money and eliminate manual upgrades which is generally considered tedious and error prone.

Most importantly, Keeping reader firmware current helps ensure users have access to the latest security and performance enhancements from ThingMagic.

Enterprise RFID Readers with Auto-Upgrade and many other features are available from ThingMagic. For more information, visit www.thingmagic.com.