Speedway® Revolution – Octane 4.8.0

Firmware Upgrade Reference Manual

For more information, contact rfid_info@impinj.com

Table of Contents

1 Introduction

The Speedway Revolution reader provides several methods for managing the firmware image which include:

- Upgrade to a new image
- Fallback to a previous valid image
- Restore to factory configuration default settings

The upgrade operations can be done without disturbing the current operation of the reader and therefore minimizes the down time required to change to a new firmware image. Restoring to a factory default configuration or falling back to a previous image both take effect immediately and force an immediate reboot to complete the commanded operation.

The user may manage the upgrade process through the steps described below. The upgrade may be performed via the command line interface or by automatic file retrieval, once it has been configured

1.1 Terms and Acronyms

The following terms and acronyms are used throughout this discussion

- Image File URI: Universal Resource Identifier for the Upgrade Image File.
- Metafile: See Upgrade Configuration Metafile.
- Metafile-URI: Universal Resource Identifier for the metafile.
- Primary image: the image that is currently running on the reader.
- Secondary image: the image that is not running and may be the target of upgrading or reserved for fallback.
- Upgrade Configuration Metafile: data file that resides on a server and contains the Upgrade Configuration information.
- Upgrade Configuration: the information used for determining the upgrading procedure.
- Upgrade Image File: the file that contains the Speedway reader image used for upgrade and is stored on a server and retrieved by the Speedway reader
- Universal Resource Identifier: URI as defined in RFC3986.

2 Dual Image Model

A flash consists of a primary and a secondary image. Each image contains three partitions as shown in Figure 2-1. The primary image is currently running on the reader, and the secondary image contains the image previously running. When an upgrade is performed, the destination is always the secondary image. After the upgrade completes and the reader reboots, the previous secondary image becomes the new primary image. The previous primary image becomes the new secondary image. As long as the secondary image contains a valid image, a fallback operation may be performed to cause the reader to revert to the previous primary image. Once an upgrade has been started, the fallback operation cannot be executed because the secondary partition no longer contains a valid image.

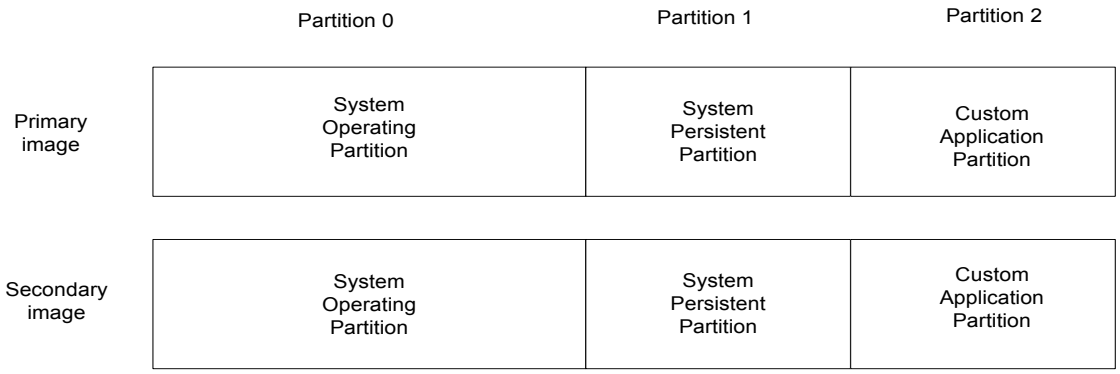

Figure 2-1 Dual Image Model

The three partitions in each image are:

- Partition 0: the System Operating Partition (SOP). This partition contains the Operating System, file system, and Impinj reader applications.
- Partition 1: the System Persistent Partition (SPP). This partition contains the Impinj reader applications configuration and other general configuration data.
- Partition 2: the Custom Application Partition (CAP). This partition may contain a custom application and its associated data.

The Dual Image Model allows an upgrade to be performed in the background. The current operation of the reader is not disturbed until the reader is rebooted. The reboot time (called the activation time) is configurable by the user.

2.1 Image Versioning Scheme

Each partition has an associated four-part version number. The version number in the upgrade configuration file is represented by a string consisting of four fields separated by "**.**" dot. See section 3.2 for details.

ddd.ddd.ddd.ddd

Each field is a decimal number ranging from 0 to 255. The left-most field is the most significant part of the version number with sub-versions provided to the right. For the purpose of upgrades, when two version numbers are compared, the one with the largest left-most number is considered a higher version and is therefore a newer image. For example, if the two versions compared are 2.3.4.9 and 2.4.4.1, then 2.4.4.1 is considered newer because the second number from the left is larger (in this case 4 versus 3). Other than this comparison, the upgrade mechanism assumes no additional meaning for the version string.

3 Upgrade Methods

Speedway provides two methods to support software upgrades: manual (known as *push* in previous versions) and auto (known as *pull* in previous versions). Manual mode is the user commanded method entered at the Rshell command line. Manual mode is used to perform an upgrade on an individual reader. Auto mode is an upgrade method, which once configured, allows a reader to:

- Periodically retrieve an Upgrade Configuration Metafile.
- Determine from the configuration data if an upgrade needs to be performed.

Auto mode also allows simultaneous upgrades of multiple readers through a single Upgrade Configuration Metafile. The default reader configuration is manual mode.

In manual mode, the user can trigger a one-time upgrade of the Speedway image. When triggering the upgrade, the user must specify the location of the Upgrade Image File as a Universal Resource Identifier (the Image File URI). The upgrade will be performed unconditionally, regardless of partition version information. Once the image downloads and programming is completed, the reader will remain in manual mode and will perform no additional upgrades until a new user request is issued. In manual mode, the reader will not reboot automatically to activate the new image. To complete the activation, the user must issue a reboot command at the Rshell command line.

✓ **Note:** A power failure is treated the same as a reboot in manual mode. This means the reader will switch to the new image after a power failure or a reboot.

In auto mode, the user creates a custom Upgrade Configuration Metafile (or Metafile). The Metafile is stored on a remote server. The user configures the location of the Metafile as a URI. The reader downloads the Metafile at a configurable periodic interval, called the retrieve period. The reader then uses the content of the Metafile to make automatic upgrade decisions. The reader remembers the retrieve mode, retrieve period, and URI across power-cycles. This allows the reader to resume the auto method after a system reboot. Typically, when the reader retrieves the Metafile, it will find that no upgrade is needed. In the absence of any change in the Metafile on the server, the image version the reader is running is the same as the version specified in the Metafile.

Following a system reboot, if the reader is configured for auto mode, the first Metafile retrieval is scheduled randomly for 3-5 minutes after reboot. The use of a delayed and random first retrieval time serves two purposes. First, in a multi-reader installation, if all units are powered on simultaneously, this will stagger the retrieve intervals across the units. The second purpose is to allow any external network services for the reader to be restored before the first Metafile retrieval occurs.

3.1 Preparing the Upgrade Image

The path and permission of the Upgrade Image File on the server should be set to properly allow file retrieval via the `upgrade-file-uri` field method specified in the Metafile. Another method is the Image File URI specified in the manual mode.

3.2 The Upgrade Configuration Metafile

The Upgrade Configuration Metafile is at the core of the auto upgrade mechanism. The user prepares this file based on upgrade requirements and saves it on a file server accessible from the reader. The Metafile contains instructions to the reader defining to how to perform the upgrade as a list of text-based entries. Each data entry consists of a single line data field and may be qualified with one or more parameters separated with a semi-colon. All data fields and parameters are mandatory unless marked as optional. The format of a data entry is:

```
field-name:field-value{;parameter-name=parameter-value}
```

All data entries in the Metafile apply to a particular reader model. The Metafile can contain entries for multiple reader models with the following entry:

```
reader-model:<string>
```

as a delimiter. No data entries come before the first `reader-model` entry. A Metafile cannot be shared between a Speedway and a Speedway Revolution Reader. The Speedway Reader does not understand the `reader-model` parameter.

Table 3-1 lists the data entries in the Metafile.

Important: The Metafile must **not** contain any Unicode characters.

Table 3-1 Upgrade Configuration Definition

Field Name	Field Value	Param Name	Param Value	Description
reader-model	**<string>**	This field is used as a delimiter. The **<string>** field value identifies which reader model the subsequent data fields are intended for. It means all data fields after this one, up to the next delimiter or end of metafile, are reader model specific. The following is a list of supported reader models: Speedway R220 Speedway R420 Speedway R640		
uc-uri (optional)	**<string>**	This field sets a URI from which the metafile is downloaded in subsequent retrievals.		
retrieve-mode	This field indicates how the metafile will be retrieved.			
	manual	This field tells the reader to wait to be given upgrade information directly.		
	auto	**retrieve-period**	**<int>**	This field tells the reader to periodically retrieve the Metafile. The mandatory parameter specifies how often (in minutes) the reader downloads the Metafile.
upgrade-mode	This field indicates how the reader determines the need to upgrade.			
	auto	The reader determines if an upgrade is necessary based on its knowledge of the local image version compared with the upgrade file. Upgrade is needed if the local image has at least one partition that has a lower version than the corresponding partition in the upgrade image file.		
	forced	The reader should upgrade as long as the current image has at least one partition that has a different version from the corresponding partition in the upgrade image file.		
commit-mode	This field indicates how the image should be activated.			
	immediate	The image activates immediately after the upgrade completes, causing an immediate reboot once programming completes.		
	wait-4-cmd	The image activates by a reboot command from the user.		
	The following parameters are defined, where time is mandatory and early-act-ok is optional. All parameters, when present, must be given in the order presented here.			
	scheduled	**time**	**<string>**	To activate the new image, reboot is scheduled at the time indicated by the mandatory parameter **time**. The value of time is a string that takes either the fully specified format "<time-zone>.yyyy:mm:dd:hh:mm:ss," or the wildcard format: "<time-zone>.*.hh:mm:ss+r<max-delay>" where <time-zone> is utc, and <max-delay> is the maximum value of a random delay. When wildcard time is used, the reboot time is the upcoming hh:mm:ss AFTER the upgrade is completed, plus a delay of random length, up to max-delay, after the hh:mm:ss. The format of max-delay is "<number>m" or "<number>s," to indicate the max delay number in minutes or seconds. See Section 3.6.7 for a detailed explanation of reboot time.
		early-act-ok (optional)	**{no, yes}**	Due to an early reboot, it is OK to activate the upgraded image before the scheduled activation time. Default value is **no** when this parameter is absent.

Table 3-2 Upgrade Configuration Definition (cont'd)

Field Name	Field Value	Param Name	Param Value	Description
dl-retries (optional)	**<int>**	Number of times to retry if download fails due to timeout. Default value is 0.		
dl-retry-period (optional)	**<int>**	Time to wait (in seconds) before retrying a download. Only applicable if **dl**-retries is non-zero. Default value is 0.		
img-type	**<int>**	This field indicates the image type of the upgrade file specified by the **upgrade-file-uri** field. The type is the enumeration number **<int>**. Refer to release notes for specific image type.		
download-mode (optional)	This field is optional and indicates when to start downloading the image file, following the Metafile retrieval. Default value is immediate when this field is absent.			
	immediate	The image file download should proceed immediately following metafile retrieval.		
	fixed-delay	**delay**	**<int>**	The image file download should be delayed **<int>** seconds following metafile retrieval.
	random-delay	**delay**	**<int>**	The image file download should be delayed for a random number of seconds up to a max of **<int>.**
upgrade-file-uri	**<string>**	This field is the URI of the upgrade image file from which the upgrade image is downloaded.		
partition	This field is the partition descriptor in an upgrade file. Refer to release notes for specific values.			
	<int>	**version**	**<string>**	Version of the partition, consisting of 4 fields of decimal numbers separated by a dot '**.**'. The number in each field must be in the range of 0 to 255.

3.3 Preparing the Upgrade Configuration Metafile

The Upgrade Configuration Metafile is prepared on the server as pointed to by the reader's Metafile URI. The data entries in the Metafile must follow the format and definition given in section 3.2. Missing mandatory data entries or improper syntax will cause the reader to reject the Metafile.

The Upgrade Image File pointed to by the `upgrade-file-uri` field must contain the same:

- partitions
- image types
- versions

as described by the **partition** fields in the Metafile. Disagreement between the Metafile and the Upgrade Image File will cause the reader to reject the downloaded image file.

The path and permission of the Metafile on the server should be set correctly to allow file retrieval via the URI parameter method specified in the `config image metafile` command of Rshell.

✓ **Note:** For a given reader model, a single `upgrade-file-uri` must be specified, limiting the upgrade to a single Upgrade Image File. Since the Upgrade Image File must contain all the partitions specified. This single file constraint guarantees a consistent compliment of partitions will be applied when the upgrade completes.

3.4 Command Line Interface

The command line interface is provided through the Rshell application. Refer to the Rshell documentation for command details. All commands provide an immediate response, indicating the command was either accepted or rejected. The set of possible responses are summarized in Table 3-3 below.

Table 3-3 Command Responses

Command Response	Meaning
`Status='0,Success'`	Command was accepted and processed.
`Status='1, Invalid-Command'`	The command was unrecognized.
`Status='2, Invalid-Command-Parameter'`	One or more of the command parameters was invalid.
`Status='3, Invalid-Parameter-Value'`	One or more of the command parameter values was invalid.
`Status='4, Parameter-Dependency-Error'`	An invalid combination of parameters was specified.
`Status='5, Incomplete-Parameter-List'`	The number of command parameters is incorrect.
`Status='8, Permission-Denied'`	The pre-conditions for the command were not satisfied.
`Status='10, Command-Being-Processed'`	A previous command or scheduled operation is being processed.

3.4.1 Upgrade

An upgrade triggers via the Command Line Interface in any one of the following scenarios:

- The user can invoke the Rshell command `config image upgrade <URI>` to instruct the reader to enter manual mode, download the upgrade image file from the specified URI, and perform an upgrade with the downloaded image. See the Rshell documentation for more details.

  Table 3-3 provides possible command responses. The following is an example of a possible error:

  A malformed URI: `Status='3, Invalid-Parameter-Value'`

- The user may invoke the Rshell command `config image metafile <URI>` to instruct the reader to enter auto mode, download a Metafile from the specified URI, and perform an upgrade based on the Metafile. Regardless of the upgrade status, the reader remembers the URI for future use.

  Table 3-3 provides possible command responses. The following is an example of a possible malformed URI error:

  ```
  Status='3, Invalid-Parameter-Value'
  ```

- The user can invoke the Rshell command, `config image retrievemode`, to set the retrieve mode of the reader. The retrieve mode settings are manual or auto. If the retrieve mode is set to auto and the reader has a valid Metafile URI, the reader will immediately attempt to retrieve the Metafile via the URI. If the Metafile retrieval fails, the reader will retry periodically based on the specified retrieve period in the command.

  Table 3-3 provides possible responses. The following are examples of possible errors when:

  - Specified auto without a period:

  ```
  Status='4, Parameter-Dependency-Error'
  ```

  - Specified auto, but a Metafile URI is not defined yet:

  ```
  Status='8, Permission-Denied'
  ```

3.4.2 Configuration Default Restore

The user returns the reader to a factory default configuration using the Rshell command `config image default`. The current primary SOP and CAP, are retained if present, but defaults the reader's configuration. The reader reboots immediately upon completion.

3.4.3 Fallback to Previous Image

The user can invoke the Rshell command `config image fallback` to restore the reader to its previous image, if a valid image is available. The reader is rebooted immediately upon completion.

Table 3-3 provides possible responses. The error indicating a valid image is not available to fallback to error displays as:

```
Status='8, Permission-Denied'
```

3.4.4 Query the Upgrade Status

The user invokes the Rshell command `show image summary` to view the details of the current primary and secondary images. This command also shows the status of pending and completed upgrades, error codes and indicates the reasons for upgrade failures. See section 3.51. **Error! Reference source not found.** for examples of typical responses. The details of each status line are described below.

Table 3-4 Status Query and meaning

Upgrade Status Query	Meaning
UpgradeStatus	The upgrade application status. Table 3-5**Error! Reference source not found.** contains the possible values.
LastOperation	Only displayed/provided in conjunction with next line (LastOperationStatus). Typically these are provided when additional information is required, for example under error scenarios or when a system reboot has been scheduled. This will be one of the values from Table 3-5**Error! Reference source not found.**, and generally reports the condition leading up to the current status.
LastOperationStatus	Provides detail description for the LastOperation. The values are given in Table 3-7**Error! Reference source not found.**.
PrimaryImageType	The image type enumeration for the primary image (refer to release notes for details
PrimaryImageState	The current state of the primary image (this should always be Active) refer to Table 3- for details of image state values
PrimaryImageSystemVersion	The current version of the primary SOP
PrimaryImageConfigVersion	The current version of the primary SPP (255.255.255.255 is the default SPP version)
PrimaryImageCustomAppVersion	The current version of the primary CAP (only displayed if CAP is present)
SecondaryImageType	The image type enumeration for the secondary image (refer to release notes for details)
SecondaryImageState	The current state of the secondary image would typically have one of the values from Table 3-
SecondaryImageSystemVersion	The current version of the secondary SOP
SecondaryImageConfigVersion	The current version of the secondary SPP (only displayed if SPP is present)
SecondaryImageCustomAppVersion	The current version of the secondary CAP (only displayed if CAP is present)

Table 3-5 Upgrade Status Values

Upgrade Status Values	Meaning
`Ready`	Upgrade application is not busy and ready for additional commands
`WaitingForMetafileTransfer`	Metafile transferring from server
`WaitingForMetafileRetry`	Metafile transfer timed out, waiting for subsequent transfer.
`ProcessingMetafile`	Metafile received and is being processed
`ExpectingGetImageReq`	System is in the download delay window (controlled by the download delay configuration setting) prior to DeterminingNeedForImageFile
`DeterminingNeedForImageFile`	Version information determining if the image file needs to be retrieved
`WaitingForImageFileTransfer`	Image file transferring from server
`WaitingForImageFileRetry`	Image file transfer timed out, waiting for subsequent transfer
`ProcessingImageFile`	Image file processing
`WaitingForCommitImage`	Image file committing to flash
`SchedulingActivation`	Image activation scheduling
`WaitingToActivateImmediate`	Image activating, and preparing for immediate reboot
`WaitingToActivateScheduled`	Image activating, and reboot is scheduled based on user specified commit time
`WaitingRandomRebootDelay`	System is in the random delay window (provided as part of commit time specification) prior to system reboot
`WaitingForFallback`	A config image fallback command is processing. Immediate system reboot when complete.
`WaitingForCDR`	A config image default command is processing. Immediate system reboot when complete.
`WaitingForRequestedReboot`	Reader is preparing rebooted.

Table 3-6 Image State Values

State Value	Meaning
Active	Image previously ran and is eligible to fallback to.
Pre-Active	Image was activated and is ready to become Primary image on next reboot.
Pending	Image was committed to flash, waiting for commit time to elapse before activating it by changing to the Pre-Active state.
Obsolete	Image was invalidated, typically due to a fallback operation.

Table 3-7 LastOperationStatus Status Strings

Status Value	Meaning
Upgrade is not required	There is no need to upgrade because the upgrade mode is auto and the current version is greater or equal to the upgrade image version.
Waiting for manual reboot	Upgrade completed and the reader waits for a manual reboot to activate the new image.
Early-Act complete	Upgrade completed and is waits for the scheduled activation time. The image activatse early if manually rebooted before the scheduled time.
Reboot/activation	Upgrade completed and waits for the scheduled activation time. The image will NOT activate early if rebooted before scheduled time.
Could not resolve host <hostname>	The given hostname could not be resolved due to a misspelled name, DNS server not configured or not reachable or other network based error.
File transfer progress timeout	File transfer timed out due to lost network connection to the file server or other network error.
Metafile validation failure, see error log for details	The downloaded metafile has errors such as wrong format, missing mandatory fields, no matching reader model etc. Details about the error are in the error log.
Failed to validate image file details with metafile	The detailed info in the image file does not match those specified in the metafile, such as image type, number of partitions and/or their versions number(s)
Failed writing body	The upgrade image size exceeds the size that the reader can accommodate.
Failed to write partition <n>	A particular partition exceeds the allowed size.

3.4.5 Query the Upgrade Configuration Settings

The user can invoke the Rshell command `show image metafile` to view the details of the current retrieve mode and Metafile data contained in the reader. The configuration settings output shows the current configuration. If a metafile is not loaded, the default settings report as follows:

```
> show image metafile
Status='0,Success'
MetafileUri=''
RetrieveMode='Manual'
RetrievePeriod='1'
UpgradeMode='Auto'
CommitMode='Immediate'
CommitTime=''
EarlyActOk='no'
DownloadRetries='0'
DownloadRetryPeriod='0'
ReaderModelName=''
ImageType='10'
DownloadMode='Immediate'
DownloadDelay='0'
ImageFileUri=''
```

If a metafile loads, then the configuration settings from the metafile are reported. Any setting changes after loading of the metafile also display, for example:

```
> show image  metafile
Status='0,Success'
MetafileUri='http://server/path/metafile.txt'
RetrieveMode='Auto'
RetrievePeriod='5'
UpgradeMode='Forced'
CommitMode='Scheduled'
CommitTime='utc.2010:01:30:23:12:00+r2m'
EarlyActOk='yes'
DownloadRetries='2'
DownloadRetryPeriod='60'
ReaderModelName='Speedway R420'
ImageType='10'
DownloadMode='Immediate'
DownloadDelay='0'
ImageFileUri='http://server/path/testing/image_28.upg'
Partition0='4.0.0.8'
>
```

✓ **Note**:

- Additional Partitions, if defined, are reported as Partition1=’xxx.xxx.xxx.xxx’, Partition2=’xxx.xxx.xxx.xxx’, etc.
- Only `RetrieveMode` and `MetafileUri` are permanently stored in the reader configuration, therefore “show image metafile” issued immediately after reboot will show default values for all but these two fields. Up-to-date values are shown after a metafile is retrieved.

The possible values for each of the settings reported are summarized in Table 3-8. The definitions of these settings are specified in Table 3-1.

Table 3-8 Configuration Setting Values

Setting	Possible Values
MetafileUri	Valid URI from config image metafile=uri command
RetrieveMode	Auto Manual
RetrievePeriod	1 <= N <= 44000
UpgradeMode	Auto Forced
CommitMode	Immediate Scheduled WaitForCommand
CommitTime	Commit time string from the metafile when commit-mode is scheduled: <time-zone>.yyyy:mm:dd:hh:mm:ss <time-zone>.*.hh:mm:ss+r<max-delay>
EarlyActOk	Yes No
DownloadRetries	1 <= N <= 5
DownloadRetryPeriod	0 <= N <= 60
ReaderModelName	Reader model string from the metafile
ImageType	10
DownloadMode	Immediate Fixed Random
DownloadDelay	0 <= N <= 360
ImageFileUri	Valid URI from config image upgrade=uri command Valid URI from upgrade-file-uri in the metafile
Partition0	000.000.000.000 <= xxx.xxx.xxx.xxx <= 255.255.255.255
Partition1	000.000.000.000 <= xxx.xxx.xxx.xxx <= 255.255.255.255
Partition2	000.000.000.000 <= xxx.xxx.xxx.xxx <= 255.255.255.255

3.4.6 Background Execution of Commands

Some `config image` commands execute in the background and take a short time to run. Until the active command finishes processing, any subsequent command will be rejected. The response code reads: `Command-Being-Processed`. Any command that initiates file retrieval or changes the image configuration results in background execution. The only `config image` command that does not result in background execution is:

```
config image retrievemode manual
```

When the `show image` command(s) completes, any other command can immediately follow.

3.5 Upgrade Examples

3.5.1 Manual Example

An example of command line activity is shown below and demonstrates a successful upgrade, using the **manual** method. The lines that begin with '>' are the Rshell commands and text entries following the number symbol, #, are comments and do not display in actual use.

```
# Issue a command to upgrade using FTP. The file path is only an
example.
>
> config image upgrade
ftp://username:password@server1.mydomain.com/binaries/sop-4_0_2_0.upg
Status=0,'Success'    # command accepted
>
> show image summary   # Query status
Status='0,Success'
UpgradeStatus='WaitingForImageFileTransfer' # image file downloading
# Current image info
PrimaryImageType='10'
PrimaryImageState='Active'
PrimaryImageSystemVersion='4.0.1.0'
PrimaryImageConfigVersion='255.255.255.255'
SecondaryImageType='10'
SecondaryImageState='Active'
SecondaryImageSystemVersion='4.0.0.0'
>
> show image summary
Status='0,Success'
UpgradeStatus=' WaitingForCommitImage' # Download OK. Erasing and
writing secondary flash
# Current image info
PrimaryImageType='10'
PrimaryImageState='Active'
PrimaryImageSystemVersion='4.0.1.0'
PrimaryImageConfigVersion='255.255.255.255'
SecondaryImageType='10'
SecondaryImageState='Active'
SecondaryImageSystemVersion='4.0.0.0'
>
> show image summary
Status='0,Success'
UpgradeStatus='Ready'
LastOperation='WaitingToActivateImmediate'
LastOperationStatus='Waiting for manual reboot'    # programming done
successfully
```

```
PrimaryImageType='10'
PrimaryImageState='Active'
PrimaryImageSystemVersion='4.0.1.0'
PrimaryImageConfigVersion='255.255.255.255'
SecondaryImageType='10'
SecondaryImageState='Pre-Active'
SecondaryImageSystemVersion='4.0.2.0'
# Reader is waiting for reboot to activate the new image. All other
activities are not affected.
>
> reboot
Status=0,'Success'
>
# when status LED comes back on as solid green, the reader will be
running from the new image
```

3.5.2 Metafile Example

The example below is a sample of a complete metafile. If a metafile contains comment lines, the comment begins with the number sign **#.** A single # denotes an alternative value or additional fields and ## denotes an explanation.

```
##  This is an example upgrade config metafile.
##  Lines commented out with single # are alternative values or
additional fields
##  Lines commented out with double ## are explanations
##
## The following settings will be applied only to R420 readers
reader-model: "Speedway R420"
##  retrieve-period is in minutes
retrieve-mode:auto;retrieve-period=60
#retrieve-mode:manual
##
## only perform the upgrade if a partition version is newer than what's
currently running
upgrade-mode:auto
#upgrade-mode:forced
##
##  reboot at a scheduled time yyyy:mm:dd:hh:mm:ss
commit-mode:scheduled;time="utc.2006:05:08:04:12:32";early-act-ok=yes
#commit-mode:wait-4-cmd
#commit-mode:immediate
##
```

```
##  dl-retries defaults to no-retry if not present.  retry only if
failed due to timeout
#dl-retries:3
##  dl-retry-period is in seconds
#dl-retry-period:60
##
img-type:10
##
## The download-mode field indicates when to start download
## absence of this field means immediate download
## when download-mode is random-delay,'delay' is the max of random
## Delay.  delay time is in seconds
#download-mode:immediate
#download-mode:fixed-delay;delay=120
download-mode:random-delay;delay=120
##
## The URI for the Upgrade Image File
upgrade-file-uri:"tftp://fileserver.store.com/R420_sop_4_0_2_0.upg"
##
##  partitions and their versions must agree with what's in the image
partition:0;version="4.0.2.0"
#partition:1;version="255.255.255.255"
#partition:2;version="1.0.0.3"
##
## The following settings will be applied only to R220 readers
reader-model: "Speedway R220"
retrieve-mode:auto;retrieve-period=30
upgrade-mode:auto
commit-mode:scheduled;time="utc.2006:05:08:10:20:00"
dl-retries:3
dl-retry-period:30
img-type:10
download-mode:immediate
upgrade-file-uri:"tftp://fileserver.store.com/R220_sop_4_0_1_0.upg"
partition:0;version="4.0.1.0"
```

3.5.3 Other Universal Resource Identifier (URI) Examples

The Speedway reader supports three URI schemes for upgrades:

- TFTP
- FTP
- HTTP

Examples of URIs:

- tftp://tftpserver.mydomain.com/image-sop-scp-cap-2.1.1.upg

- ftp://user:password@ftpserver.mydomain.com/speedway/images/image-sop-scp-cap-2.1.1.upg
- http://httpserver.mydomain.com/impinj/reader-images/upgrade_metafile

As with any remote file retrieval, the servers should be properly configured such that the files are accessible either anonymously or by the specified user from the client-reader.

3.6 Detailed Upgrade Behavior

3.6.1 Upgrade File Validity Check

The reader always checks the validity of the upgrade file by checking the:

- Upgrade file format
- Upgrade file CRC(s)
- Hardware compatibility with the reader
- Product type compatibility with the reader
- Agreement between the upgrade metafile and the upgrade image in terms of version number, image type and partitions present.

If the check fails, the upgrade is aborted and the status is reported via the `show image summary` Rshell command.

3.6.2 Rapid Polling Intervals

If the reader is configured to update automatically, the `retrievemode` is auto. If the user attempts a `config image` command at the same time the automatic update occurs, it is possible that the user will receive the `Command-Being-Processed` message. This situation will most likely occur only if the user's network is: slow, heavily loaded, and the retrieve period (polling interval) is short.

3.6.3 Upgrade Decision

Several factors influence a successful upgrade and not all upgrade attempts will result in an actual upgrade, even when the upgrade file is valid. The reader's upgrade decision is based on the following factors:

- Image versions of the SOP and CAP partitions of the primary image.
- Image version(s) of the partition(s) in the Upgrade Configuration Metafile
- Upgrade Image File downloaded, including the number of partitions present
- Image type of the primary image, and the type indicated by both the Upgrade Configuration Metafile and Upgrade Image File.
- Upgrade mode, **forced** or **auto**, as indicated in the Upgrade Configuration Metafile.

In **auto** upgrade mode, the upgrade will occur only when either one of the following is true:

- The upgrade image has the same type as the primary image and at least one partition in the upgrade image has a version higher than the corresponding version in primary image. The current primary partition image will be retained when the partition in the upgrade file has a lower version number than the current primary image.
- The upgrade image has a different image type from the primary and SOP is present in the upgrade file.

In **forced** upgrade mode, the upgrade will occur only when either of the following is true:

- The upgrade image has the same type as the primary image and at least one partition in the upgrade file has a different version than the primary image.

- The upgrade image has a different image type than the primary and the SOP is present in the upgrade file.

✓ **Note:** For any case where a partition downgrade is required, the **forced** upgrade mode must be specified. Without the **forced** upgrade, the application will not upgrade to the specified firmware image.

If the `config image upgrade` command is used, the upgrade **always** performs regardless of version numbers or image type.

3.6.4 Download Retry Behavior

As indicated in Table 3-2, the download of a metafile or upgrade image is retried if it fails due to a timeout. The retry wait time specified by `dl-retry-period` is the wait time in addition to the time it takes for the upgrade agent to detect a failure. The upgrade agent typically measures 30 seconds of inactivity in download before declaring a failure. As such, the setting of 5 retries and 10 sec of retry wait time would lead to a retry process that lasts 5*(10+30)=200 sec if timeout failure persists.

When the next scheduled metafile retrieval is due, an unfinished download retry from a previous retrieval is aborted. As such it is suggested that `dl-retries` and `dl-retry-period`**,** if used**,** be set to make the retry process short relative to `retrieve-period` in order to avoid unnecessarily retries between scheduled retrievals.

3.6.5 Partition Copy-Over

There are times in which the Upgrade Image File does not necessarily contain all the partitions. In these cases, provided the image type is the same as the current primary image, the missing partition(s) will be copied over to the secondary image from the primary image as required. The behavior is as follows:

- If the Upgrade Image File contains a SOP only, the primary SPP and CAP (if present) are copied over.
- If the Upgrade Image File contains a SOP and a CAP, the primary SPP is copied over.
- If the Upgrade Image File contains just a CAP, the primary SOP and the SPP are copied over.
- If the upgrade file contains a SOP and SPP, the primary CAP, if present, is copied over.

3.6.6 Image partitions already programmed

Depending on the configuration in the Metafile it is possible the partitions in the Upgrade Image File are already on the secondary image. For example if the retrieve period for the Metafile is ten minutes and a reboot is scheduled in ten hours. Following a successful upgrade the reader will still retrieve the Metafile every ten minutes. Provided the image type is the same as the current secondary image, and since all of the intended partitions are already programmed, no

reprogramming will take place. It is also possible for the Metafile to be changed before the reboot in which case the upgrade will be performed again with the new data.

This behavior only applies to automatic upgrades performed via the periodic auto `retrievemode` method with an upgrade mode of auto or forced. When the upgrade is manually commanded with the `config image upgrade` command, the flash memory is always programmed with the upgrade image regardless of the versions on the primary and secondary images.

3.6.7 Scheduled activation of new image

3.6.7.1 Schedule activation time

When `commit-mode` is set to scheduled, a reboot time must be specified using the `time` parameter. See Table 3-1 for details. There are two formats for specifying time; the fully specified format:

```
utc.yyyy:mm:dd:hh:mm:ss
```

and the wildcard format:

```
utc.*.hh:mm:ss
```

There is a caveat in the use of wildcard time. Since the `hh:mm:ss` field of the reboot in the wildcard time is relative to the time of the upgrade completion, the actual reboot time may depend on when the upgrade is completed during the day. For example, if the wildcard reboot time is 23:00:00 and the upgrade is completed by 16:00:00, the reboot is 7 hours away. But if the upgrade is completed by 23:30:00, the reboot will be 23.5 hours away (i.e., at 23:00:00 the next day).

When specifying the desired wildcard reboot time in the Metafile, two delay factors should be considered:

1. the time it takes for the reader to check the Metafile (for example: the retrieve-period);
2. the time it takes to perform the upgrade.

Always modify the Metafile well ahead of the intended wildcard reboot time if a same-day reboot is desired.

There is another caveat when scheduled commit-mode is used and an early activation is not intended. In this scenario, if a reboot occurs before the scheduled activation time, the reader will still run the old image, but it will not know the original activation time until the metafile is successfully retrieved. This means a scheduled activation of a new image will not occur if the reader is rebooted and then fails to retrieve the metafile due to a disrupted network connection for example.

3.6.7.2 Change of metafile before scheduled activation

When the activation of a new image is scheduled in the future, the reader will periodically retrieve the metafile and act on it. It is possible to change the metafile and to modify the upgrade

behavior if the change is made early enough. For example the following can be done by changing the metafile:

- Upgrade to a different image
- Change the activation time
- Change whether an early activation is OK

✓ **Note:** Change is only possible if the metafile is modified at least one retrieve-period before the schedule activation time.

3.6.8 Change of metafile URI in the metafile

An optional data field `uc-uri` can be used in the metafile to specify a metafile URI from which the metafile will be downloaded in future retrievals. See Table 3-1. As an alternative to the `config image metafile <metafile-uri>` command, the `uc-uri` field provides a way of re-directing the reader to a new metafile location using the current location metafile. The new metafile URI will be saved to persist across reader reboots.

As with any change in the metafile, the modification should be made at least one retrieve-period before the intended time for it to take effect.

4 Custom Application Upgrade

The custom application (CAP) is notified if a CAP is present in the current image. This allows the CAP to complete any required upgrade related actions to the following two events:

- Image upgrade
- Configuration Default Restore (CDR)

The method for the custom app to receive the notification is to have an executable program at

```
/cust/cust_app_upgrade
```

On the first boot after a CDR, `cust_app_upgrade` is called as follows:

```
/cust/cust_app_upgrade cdr
```

This notifies the custom app that the reader was restored to its default configuration. The custom application should also restore any configuration necessary.

On the first boot after an image upgrade and assuming a CAP exists on the secondary image, `cust_app_upgrade` is called as follows:

```
/cust/cust_app_upgrade upg <cust_dir> <old_cust_dir>
```

where `cust_dir` is the root of the custom app directory,for example `/cust,` and `<old_cust_dir>` is the previous custom app root directory. This notifies the new custom app that an upgrade occurred and the custom application can access its old directory to import configuration or other data needed.

In addition to the command line arguments, the following environment variables with exemplary values are exported to the `cust_app_upgrade` program:

```
primary_sop_vsn=4.0.0.7
primary_cap_vsn=1.0.1.0
secondary_sop_vsn=4.0.0.7
secondary_cap_vsn=1.0.2.0
```

where `primary_sop_vsn/primary_sop_vsn` are the versions of the current SOP/CAP and `secondary_sop_vsn/secondary_cap_vsn` are the previous versions of SOP/CAP if one exists.

The following restrictions apply to the `cust_app_upgrade` program:

- When `cust_app_upgrade` is called, the reader has not completed its boot sequence. For example, no RFID application is running and the network is not set up. As such, this program should not start the full custom application or attempt to retrieve data from the network.
- When `cust_app_upgrade` is called after an upgrade, the `old_cust_dir` is a temporary read-only directory on which the previous custom app is mounted and it is no longer accessible after the `cust_app_upgrade` terminates.
- The `cust_app_upgrade` is called as part of the initial reader startup sequence and it runs finitely; otherwise the rest of the system delays and, if it delays for more than several minutes, it causes the reader to reboot.

A Bash program example `cust_app_upgrade` is:

```
#!/bin/sh
event=$1
echo “cust upgrade script enters”
echo “my SOP version is $primary_sop_vsn”
echo “my CAP version is $primary_cap_vsn”
test ! -z $secondary_sop_vsn  && echo “There is an old SOP of version
$secondary_sop_vsn”
test ! -z $secondary_cap_vsn  && echo “There is an old CAP of version
$secondary_cap_vsn”

if [ $event = “cdr” ] ; then
  echo “Reader restored its default configuration”
  # do something
elif [ $event = “upg” ] ; then
  echo “Reader just had an upgrade”
  echo “My old CAP version is $secondary_cap_vsn”
  cust_dir=$2
  old_cust_dir=$3
# copy some config from old app
  if [ -f $old_cust_dir/my_config ] ; then
    cp $old_cust_dir/my_config $cust_dir/my_old_config
  fi
fi
```

5 Revision History

Date	Revision	Comments
04/01/2009	1.0	Original release
04/15/2009	1.1	Added ExpectingGetImageReq Added clarification for scheduled activation behavior when metafile is not unavailable on reboot
04/24/2009	1.2	Update formating
08/27/2009	4.2	Changed incorrect strings for ‘auto’ and ‘manual’ modes. Push/pull are no longer reported. Added a NOTE on metafile info displayed immediately after a reboot being different from that subsequent to a metafile retrieval. Added a sub-section on detailed download retry behavior. Added uc-uri in the metafile entry table and a sub-section on its use. Added a table for lastOperationStatus strings Finalized for release
04/12/2010	4.4	Update revision and copyrights for Octane 4.4.
10/27/2010	4.6	Update revision for Octane 4.6
04/25/2011	4.8	Update revision for Octane 4.8, no changes

Notices:

The information contained in this document is confidential and proprietary to Impinj, Inc. This document is conditionally issued, and neither receipt nor possession hereof confers or transfers any right in, or license to, use the subject matter of any drawings, design, or technical information contained herein, nor any right to reproduce or disclose any part of the contents hereof, without the prior written consent of Impinj and the authorized recipient hereof.

Impinj reserves the right to change its products and services at any time without notice.

Impinj assumes no responsibility for customer product design or for infringement of patents and/or the rights of third parties, which may result from assistance provided by Impinj. No representation of warranty is given and no liability is assumed by Impinj with respect to accuracy or use of such information.

These products are not designed for use in life support appliances, devices, or systems where malfunction can reasonably be expected to result in personal injury.