Firmware Upgrade Reference Manual

version 7.3

Contents

1 Introduction

Impinj Speedway Reader and xPortal, xSpan and xArray Gateways provide several methods for managing the embedded firmware image, including:

- upgrade to a new image
- fallback to a previous valid image
- restore to factory configuration default settings

Upgrade operations can be completed without disturbing the current operation of the Reader, which minimizes the downtime 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 requested operation.

You can manage the upgrade process by using the procedures described in this document. After the upgrade process has been configured, you can perform the upgrade by using the command line interface, or by automatic file retrieval.

1.1 Terms and Acronyms

In this document, the term **Reader** is used to refer to an Impinj Speedway Reader, xPortal Gateway, xSpan Gateway and an xArray Gateway. If the section refers to only one specific device, the device name is used.

The following terms and acronyms are used throughout this document.

- **Image File URI**: Universal Resource Identifier for the Upgrade Image File.
- **Metafile**: See “Upgrade Configuration Metafile”.
- **Metafile-URI**: The 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 can be the target of an upgrade, 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 upgrade procedure.
- **Upgrade Image File**: The file that contains the Reader image used for upgrade. It is stored on a server and retrieved by the 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 typically 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. After an upgrade has been started, the fallback operation cannot be executed because the secondary partition no longer contains a valid image.

	Partition 0	Partition 1	Partition 2
Primary Image	System Operating Partition	System Persistent Partition	Custom Application Partition
Secondary Image	System Operating Partition	System Persistent Partition	Custom Application Partition

Figure 2.1 Dual Image Model

There are three partitions in each primary and secondary image:

- Partition 0: the System Operating Partition (SOP). This partition contains the operating system, root file system, and Impinj Reader applications.
- Partition 1: the System Persistent Partition (SPP). This partition contains the configuration for the Impinj Reader applications, as well as 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 ".", as shown in the example below. For more information, see section 3.2.

Example: *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

The Reader provides two methods to support software upgrades: **manual** and **auto**. In previous versions, the **manual** method was known as *push* and the **auto** method was known as *pull*. The default Reader configuration is manual mode.

Manual mode

Manual mode is started by a user command; the user enters this mode at the RShell command line. Use manual mode to perform a one-time upgrade of an individual Reader.

As part of the process of starting the upgrade, the user must specify the location of the Upgrade Image File as a Universal Resource Identifier (the Image File URI). The upgrade is performed unconditionally, regardless of partition version information.

After the upgrade image downloads and the process completes, the Reader remains in manual mode and will not perform additional upgrades until a new user request is issued. In manual mode, the Reader does not reboot automatically to activate the new image. To complete the activation, the user must issue a reboot command at the RShell command line.

Upgrading the Reader's firmware using the web page is another way of initiating an manual upgrade.

Note: A power failure is treated the same as a reboot in manual mode. This means the Reader switches to the new image after a power failure or a reboot.

Auto mode

Auto mode is an upgrade method that, after it has been 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 the use of a single Upgrade Configuration Metafile.

In auto mode, the user creates a custom Upgrade Configuration Metafile ("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 finds that no upgrade is needed. In the absence of any Metafile changes on the server, the Reader image version that is running and the Metafile versions are the same.

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 staggers the retrieve intervals across the units. In addition, the delay provides time for any network services that are external to the Reader to be restored before the first Metafile retrieval occurs.

3.1 Preparing the Upgrade Image

Set the path and permission of the Upgrade Image File on the server so it properly allows file retrieval via the upgrade-file-uri field method that is specified in the Metafile. Another method is to use the Image File URI that you specify in manual mode.

3.2 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 accessable to the Reader. The Metafile contains instructions in a list of text-based entries that tell the Reader how to perform the upgrade. Each data entry consists of a single-line data field and can be qualified with one or more parameters separated with a semi-colon. All data fields and parameters are mandatory unless indicated 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 as a delimiter:

reader-model:<string>

No data entries can come before the first reader-model entry.

A Metafile cannot be shared between an original Speedway Reader and a Speedway Reader, xPortal, xSpan or xArray Gateway. The original Speedway Reader does not understand the **reader-model** parameter. The next section lists the data entries in the Metafile.

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

3.2.1 Upgrade Configuration Definitions

reader-model

- Field Value: <string>
  - Description: This field is used as a delimiter. The <string> field value identifies the Reader model for which the subsequent data fields are intended for. It means that all data fields after this one, up to the next delimiter or end of metafile, are specific to this Reader model. The following is a list of supported Reader models:
    - Speedway R120
    - Speedway R220
    - Speedway R420
    - Speedway R640 (xPortal)
    - Speedway R660 (xSpan)
    - Speedway R680 (xArray)

uc-uri (optional)

- Field Value: <string>
  - Description: 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.

- Field Value: **manual**
  - Description: This field tells the Reader to wait to be given upgrade information directly.
- Field Value: **auto**
  - Param Name: retrieve-period
  - Param Value: <int>
  - Param Description: 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.

- Field Value: **auto**

  - Description: The Reader determines if an upgrade is necessary based on its knowledge of the local image version compared with the upgrade file. An upgrade is required if the local image has at least one partition that has a *lower* version than the corresponding partition in the upgrade image file.
- Field Value: **forced**
  - Description: The Reader determines if an upgrade is necessary based on its knowledge of the local image version compared with the upgrade file. An upgrade is required if the local image has at least one partition that has a *different* version than the corresponding partition in the upgrade image file.

commit-mode

This field indicates how the image should be activated.

- Field Value: **immediate**
  - Description: The image activates immediately after the upgrade completes, and causes an immediate reboot after the activation completes.
- Field Value: **wait-4-cmd**
  - Description: The image activates through a reboot command from the user.

For the next parameters, **time** is mandatory and **early-act-ok** is optional. All parameters, when present, must be provided in the order presented here.

- Field Value: **scheduled**
  - Param Name: **time**
  - Param Value: <string>
  - Param Description: To activate the new image, reboot is scheduled at the time indicated by the mandatory parameter time. The value of the time parameter is a string that is in one of two formats, either the fully specified format of:

    "<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', which indicates the max delay number in minutes or seconds. See Section 3.6.7 for a detailed explanation of reboot time.

  - Param Name: **early-act-ok** (Optional)
  - Param Value: {no, yes}
  - Param Description: Specifies whether it is OK to activate the upgraded image before the scheduled activation time, due to an early reboot. When this parameter is absent, the default value is **no**.

dl-retries (Optional)

- Field Value: <int>
  - Description: Number of times to retry if the download fails due to timeout. The default value is 0.

dl-retry-period (Optional)

- Field Value: <int>
  - Description: Time to wait (in seconds) before retrying a download. Only applicable if dl-retries is non-zero. The default value is 0 (immediate).

img-type

- Field Value: <int>
  - Description: 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 the release notes for the specific image type. To date, only type 10 is supported.

download-mode (optional)

This field is optional and indicates when to start downloading the image file, following the Metafile retrieval. The default value is **immediate** when this field is absent.

- Field Value: **immediate**
  - Description: The image file download should proceed immediately after the metafile retrieval.
- Field Value: **fixed-delay**
  - Param Name: delay
  - Param Value: <int>

    - Param Description: The image file download should be delayed <int> seconds following the metafile retrieval.

- Field Value: **random-delay**
    - Param Name: delay
    - Param Value: <int>
    - Param Description: The image file download should be delayed for a random number of seconds, up to a max of <int>.

upgrade-file-uri

- Field Value: <string>
    - Description: 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. The Octane firmware is partition 0, CAP firmware is partition 2. The partition version is specified when the upgrade file is created.

- Field Value: <int>
    - Param Name: version
    - Param Value: <string>
    - Param Description: The 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 Metafile URI for the Reader points to the Upgrade Configuration Metafile, which is prepared on the server. The data entries in the Metafile must follow the format and definitions provided in section 3.2.

Note: Missing mandatory data entries or improper syntax will cause the Reader to reject the Metafile.

The Upgrade Image File referenced 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 permissions of the Metafile on the server must be set correctly. This allows for file retrieval using the URI parameter method that is specified in the RShell command **config image metafile**.

Note: For a given Reader model, a single **upgrade-file-uri** field must be specified. This limits the upgrade to a single Upgrade Image File, since the Upgrade Image File must contain all the partitions specified. This single file constraint guarantees that a consistent compliment of partitions will be applied when the upgrade completes.

3.4 Command Line Interface

The RShell application provides access to the command line interface. Refer to the *RShell Reference Manual* for details about RShell commands.

All commands provide an immediate response, which indicates whether the command was accepted or rejected. The set of possible responses are summarized in Table 3.1.

Table 3.1 Command Responses

Command Response	**Meaning**
Status='0,Success'	Command was accepted and processed.
Status='1, Invalid-Command'	The command was not recognized.
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 is triggered via the Command Line Interface in any one of the following scenarios:

- Use 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 Reference Manual* for more details.

  Table 3.1 provides possible command responses. The following is an example of a possible malformed URI error:

  Status='3, Invalid-Parameter-Value'

- Use 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 3.1 provides possible command responses.

- Use 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 immediately attempts to retrieve the Metafile via the URI. If the Metafile retrieval fails, the Reader reattemps the retrieval periodically, based on the specified retrieve period in the command.

  Table 3.1 provides possible responses. The following are examples of possible errors:

  - **Auto** is specified without a period:

    Status='4, Parameter-Dependency-Error'

  - **Auto** is specified, but a Metafile URI is not defined yet:

    Status='8, Permission-Denied'

3.4.2 Configuration Default Restore

Use the RShell command **config image default** to return the Reader to a factory default configuration. The current primary SOP and CAP are retained if they are present, but the configuration of the Reader returns to default values. The Reader reboots immediately upon completion.

3.4.3 Fallback to Previous Image

Use the RShell command **config image fallback** to restore the Reader to its previous image when a valid image is available. The Reader immediately reboots upon completion.

Table 3.1 provides possible responses. The following error indicates that a valid image is not available for the fallback process:

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

3.4.4 Query the Upgrade Status

Use 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, displays error codes, and indicates the reasons for upgrade failures. See section 3.5.1. for examples of typical responses. The details of each status line are described in Table 3.2.

Table 3.2 Status Query and meaning

Upgrade Status Query	**Meaning**
UpgradeStatus	The upgrade application status. See Table 3.3 for the possible values.
LastOperation	This status is only displayed/provided in conjunction with the next status (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 3.3 and generally reports the condition leading up to the current status.
LastOperationStatus	Provides a detailed description for the LastOperation. Valid values are provided in Table 3.5.
PrimaryImageType	The image type enumeration for the primary image. Refer to the release notes for details.
PrimaryImageState	The current state of the primary image. This should always be **Active**. Refer to Table 3.4 for the meaning 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. This displays only if CAP is present.

Upgrade Status Query	Meaning
SecondaryImageType	The image type enumeration for the secondary image. Refer to the release notes for details.
SecondaryImageState	The current state of the secondary image would typically have one of the values from Table 3.4.
SecondaryImageSystemVersion	The current version of the secondary SOP.
SecondaryImageConfigVersion	The current version of the secondary SPP and only displays if SPP is present.
SecondaryImageCustomAppVersion	The current version of the secondary CAP and only displays if CAP is present.

Table 3.3 Upgrade Status Values

Upgrade Status Values	Meaning
Ready	Upgrade application is not busy and is ready for additional commands.
WaitingForMetafile - Transfer	Metafile transferring from server.
WaitingForMetafile - Retry	Metafile transfer timed out, waiting for subsequent transfer.
ProcessingMetafile	Metafile received and is being processed.
ExpectingGetImage - Req	System is in the download delay window (controlled by the download delay configuration setting) prior to DeterminingNeedForImageFile.
DeterminingNeedFor - ImageFile	Version information that determines if the image file needs to be retrieved.
WaitingForImageFile - Transfer	Image file transferring from server.
WaitingForImageFile - Retry	Image file transfer timed out; waiting for subsequent transfer.
ProcessingImageFile	Image file processing.
WaitingForCommit - Image	Image file committing to flash.
SchedulingActivation	Image activation scheduling.
WaitingToActivate - Immediate	Image activating, and preparing for immediate reboot.
WaitingToActivate - Scheduled	Image activating, and reboot is scheduled based on user-specified commit time.
WaitingRandom - RebootDelay	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.

Upgrade Status Values	Meaning
WaitingForRequested - Reboot	Reader is preparing to reboot.

Table 3.4 Image State Values

State Value	Meaning
Active	Image previously ran and is eligible as fallback image.
Pre-Active	Image was activated and is ready to become Primary image on next reboot.
Pending	Image was committed to flash, and is 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 3.5 LastOperationStatus Status Strings

Status Value	Meaning
Upgrade is not required	There is no need to upgrade because the upgrade mode is a and the current version is greater or equal to the upgrade i version.
Waiting for manual reboot	Upgrade completed and the Reader waits for a manual rebo activate the new image.
Early-Act complete Reboot <time> (excl. rand)	Upgrade completed and is waiting for the scheduled activation time. The image activates early if manually rebooted before the scheduled time.
Reboot/activation	Upgrade completed and is waiting for the scheduled activat time. The image will NOT activate early if rebooted before scheduled time.
Could not resolve host <hostname>	The given hostname could not be resolved because of a misspelled name, the DNS server not being configured or not reachable, or other network-based error.
Image file transfer failed [, scheduled retry]	File transfer timed out due to lost network connection to the file server or other network error. If specified transfer will be re-tried.
Metafile URI is not valid	The metafile URI must be configured before the retrieve mo set to auto
Metafile transfer timed out, scheduled retry	Metafile transfer timed out due to lost network connection to the file server or other network error. The transfer will be re-tried.

Status Value	**Meaning**
Unable to retrieve metafile	Metafile retrieval failed for some reason other than a timeou
Metafile validation failure. See error log for details	The downloaded metafile has errors such as wrong format, missing mandatory fields, no matching Reader model, and so forth. Details about the error are in the error log.
Unable to copy primary CAP. See error log for details	This would occur if the primary image's CAP has out-grown the (secondary) partition into which it would have been copied. The upgrade is aborted, Details about the error are in the error log.
No Matching Hardware Version.	The downloaded upgrade file isn't appropriate with this product's hardware revision. Check the firmware release no
No Matching Product Type.	The downloaded upgrade file isn't appropriate with this pro Check the firmware release notes.
Image file CAP too big See error log for details.	The downloaded image's CAP partition is too large to fit in the designated Flash memory partition. Details about the error are in the error log.
Failed to parse image file	The detailed info in the image is invalid, possibly corrupted
Failed to validate image file details with metafile	The detailed info in the image file does not match the information specified in the metafile, such as image type, number of partitions, and/or their versions number(s).
Authentication failure	Login credentials or keys necessary to complete the upgrade either not present or incorrect.

3.4.5 Query the Upgrade Configuration Settings

Use the RShell command **show image metafile** to view the details of the current retrieve mode, and the Metafile data contained in the Reader. The output from this command shows the current configuration settings.

If a metafile is not loaded, the default settings report looks like this:

```
> 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 settings that were changed after the metafile loads 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. This means that if the "show image metafile" command is issued immediately after a reboot, it 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 3.6. The definitions for these settings are provided in Section 3.2.1.

Table 3.6 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. If you attempt to run another command while these commands are running, it will be rejected until the currently active command finishes processing. 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

This section provides two upgrade examples:

- An example of command line activity for a sample upgrade that uses the **manual** method.
- An example of a complete metafile that might be used for the **auto** upgrade method.

3.5.1 Manual Upgrade Example

This section provides an example of command line activity for a successful upgrade that uses the **manual** method. The lines that begin with '>' are the RShell commands.

Issue a command to upgrade using SFTP. The file path shown is only an example.

Initiate the firmware upgrade

```
> config image upgrade
sftp://username:password@server1.mydomain.com/binaries/sop-4\_0\_2\_0.upg
Status=0,'Success' # command accepted
```

Query the status of the current firmware images

```
> show image summary
Status='0,Success'
UpgradeStatus='WaitingForImageFileTransfer'
PrimaryImageType='10'
PrimaryImageState='Active'
PrimaryImageSystemVersion='4.0.1.0'
PrimaryImageConfigVersion='255.255.255.255'
SecondaryImageType='10'
SecondaryImageState='Active'
SecondaryImageSystemVersion='4.0.0.0'
```

Confirms that the image file is downloading

```
> show image summary
Status='0,Success'
UpgradeStatus=' WaitingForCommitImage'
# 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'
```

Confirms the download was successful and is now writing to the secondary flash image.

```
> show image summary
Status='0,Success'
UpgradeStatus='Ready'
LastOperation='WaitingToActivateImmediate'
LastOperationStatus='Waiting for manual reboot'
PrimaryImageType='10'
PrimaryImageState='Active'
PrimaryImageSystemVersion='4.0.1.0'
PrimaryImageConfigVersion='255.255.255.255'
SecondaryImageType='10'
SecondaryImageState='Pre-Active'
SecondaryImageSystemVersion='4.0.2.0'
```

Confirms the secondary image flash programming completed successfully. The Reader is now 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 Auto Upgrade Metafile Example

This section provides an example of a complete metafile that might be used for a successful upgrade that uses the **auto** method. In a metafile, a 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, in this case Speedway R420s are handled differently to Speedway R220s.

```
##
##   Example Upgrade Configuration Metafile
##
## Lines commented out with double ## are explanations
## Lines commented out with single #  are alternatives values and fields
##

## The following settings apply only to R420 Readers
reader-model: "Speedway R420"

## retrieve-period is in minutes
retrieve-mode:auto;retrieve-period=60
#retrieve-mode:manual

## Perform the upgrade if a partition version is newer than 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

## Download retries only performed if failed due to timeout
## 'dl-retries' defaults to 0 (no-retry) if not present.
#dl-retries:3
## dl-retry-period is in seconds
#dl-retry-period:60

## The upgrade image is type 10
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:random-delay;delay=120
#download-mode:immediate
#download-mode:fixed-delay;delay=120

## The URI for the Upgrade Image File
upgrade-file-uri:"sftp://fileserver.store.com/octane_4_0_2_0.upg"

## partitions and their versions must agree with the image content
partition:0;version="4.0.2.0"
#partition:2;version="1.0.0.3"

## The following settings apply only to R220 Readers (w/o comments)
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:"sftp://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 Reader supports two URI schemes for upgrades:

- SFTP (FTP over SSH)
- HTTP

Examples of URIs:

```
sftp://user:password@sftpserver.mydomain.com/speedway/images/octane-5.4.0.upg

sftp://sftpserver.mydomain.com/speedway/images/mycap_v2.04.upg

http://httpserver.mydomain.com/impinj/reader-images/upgrade_metafile
```

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

When using SFTP URIs, the SFTP server login credentials can be stored (encrypted) on the Reader, captured from the command line. Once provisioned the login credentials may be omitted from the URI, in which case the Reader will directly authenticate with the SFTP server.

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 following:

- 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 by using the RShell command **show image summary**.

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 that the automatic update occurs, it is possible that the user will receive the **Command-Being-Processed** message. This situation most likely occurs only if the user's network is slow, heavily loaded, or if 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 decision to upgrade by the Reader 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
- the Upgrade Image File that was 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
- the upgrade mode, either **auto** or **forced**, as indicated in the Upgrade Configuration Metafile

In **auto** upgrade mode, the upgrade occurs only when 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 the primary image. If the partition in the upgrade file has a lower version number than the current primary image, the current primary partition image is retained.
- The upgrade image has a different image type from the primary image, and the SOP is present in the upgrade file.

In **forced** upgrade mode, the upgrade occurs only when 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 file has a different version than the primary image.
- The upgrade image has a different image type than the primary image, 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 described in Section 3.2, the download of a metafile or upgrade image is retried if it fails due to a timeout. The retry wait time that is 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 during a download before declaring a failure. For example, setting the retry wait time to 5 retries and 10 seconds of retry wait time would lead to a retry process that lasts 5*(10+30)=200 seconds if timeout failure persists.

When the next scheduled metafile retrieval is due, an unfinished download retry from a previous retrieval is aborted. Therefore, you might want to set **dl-retries** and **dl-retry-period**, if used, to a value that makes the retry process short relative to **retrieve-period**. This can help 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, as long as 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 an SOP only, the primary SPP and CAP (if present) are copied over.
- If the Upgrade Image File contains an 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 Image File contains an SOP and a 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 that 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, then, following a successful upgrade, the Reader will retrieve the Metafile every ten minutes. Because all of the intended partitions are already programmed, no reprogramming will take place if the image type is the same as the current secondary image. However, if the Metafile is changed before the reboot, the upgrade is performed again with the new data.

This behavior only applies to automatic upgrades performed by using the periodic **auto retrieve-mode** method that have 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 the new image

When **commit-mode** is set to **scheduled**, you must specify a reboot time by using the **time** parameter. See Section 3.2.1 for configuration details. There are two formats for specifying time:

Fully specified format:

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

Wildcard format:

utc.*.hh:mm:ss

> **Note:** Be aware that, when you use wildcard time, the actual reboot time might depend on when the upgrade is completed during the day. This is because the hh:mm:ss field of the reboot in the wildcard time is relative to the time of the upgrade completion.
>
> 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 you specify the desired wildcard reboot time in the Metafile, two delay factors should also be considered:

- the time it takes for the Reader to check the Metafile (for example: the **retrieve-period**)
- the time it takes to perform the upgrade

If you want a same-day reboot, be sure to modify the Metafile well ahead of the intended wildcard reboot time.

Note: Be aware of a possible issue that could occur 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 after the metafile is successfully retrieved. This means that the scheduled activation of a new image will not occur if the Reader is rebooted, and then fails to retrieve the metafile due to an issue such as a disrupted network connection, for example.

3.6.8 Change of metafile before scheduled activation

When the activation of a new image is scheduled for a future time, 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 before the scheduled activation time. For example, you can change the metafile to do the following:

- upgrade to a different image
- change the activation time
- change whether an early activation is OK

Note: These changes are only possible if the metafile is modified at least one retrieve-period before the schedule activation time.

3.6.9 Change of metafile URI in the metafile

The optional data field **uc-uri** can be used in the metafile to specify a metafile URI from which the metafile can be downloaded in future retrievals. For more information about this data field, see Section 3.2.1.

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 by using the current location metafile. The new metafile URI is saved and persists across Reader reboots.

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

3.6.10 Upgrade size constraints

If you are developing on-reader embedded applications, you should be aware of the embedded Flash memory limitations. These storage limitations are dependent on the Printed Circuit Board Assembly (PCBA) revision of the Reader. This revision is visible on the label attached to Readers, and can be determined programmatically on any Reader by using the RShell command **show system platform**.

Typically a user developing an on-reader custom application will be creating a Custom Application Partition (CAP) upgrade file. A CAP upgrade can be combined with another partition by literally concatenating it with other upgrade file(s).

For example, a CAP can be combined with an Octane firmware System Operating Partition (SOP) upgrade file. The resulting upgrade file can be used to upgrade both the SOP and CAP at the same time. The Reader validates each partition included in the combined upgrade file before accepting the upgrade.

A valid individual or combined upgrade file must conform to the size limitations shown in Table 3.7.

Table 3.7 Upgrade file size limitations

Parameter (Maximum)	**PCBA v4.xx (and lower)**	**PCBA v5.xx (and higher)**
Overall upgrade (.upg) file size	32 Mbytes	63 Mbytes
System Operating Partition (SOP)	16 Mbytes	28 Mbytes
System Persistent Partition (SPP)	8 Mbytes	16 Mbytes
Custom Application Partition (CAP)	8 Mbytes	32 Mbytes

4 Custom Application Upgrades

The embedded custom application is notified of an upgrade if a Custom Application Partition (CAP) is present in the current image. These 'hooks' allow the custom application to complete any required actions related to the upgrade, specifically the following two events:

- Configuration Default Restore (CDR)
- Image upgrade

If these events are of interest to the custom application, an executable program (or script) must named and located as follows:

```
/cust/cust_app_upgrade
```

Following either upgrade event, this executable will be called.

Configuration Defaults Restore (CDR)

On the first boot after a Configuration Default Restore (CDR) the **cust_app_upgrade** executable will be called and passed a single **cdr** parameter as follows:

```
/cust/cust_app_upgrade cdr
```

This notifies the custom application that the Reader was restored to its default configuration.

Image Upgrade

On the first boot after an image upgrade, assuming a CAP exists on the secondary image, cust_app_upgrade is called using this command:

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

where <cust_dir> is the root of the (new) primary custom application directory and <old_cust_dir> is the (former primary) now secondary custom application directory. This may be useful if custom application wants to carry-over some configuration or state from the old CAP partition into the new CAP partition.

Environment Variables

In addition to the command line arguments, the following environment variables, shown 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_cap_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.

Restrictions to the custom application upgrade program

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. Therefore, 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 that the previous custom application has mounted. This temporary directory is not accessible after the **cust_app_upgrade** program terminates.
- The **cust_app_upgrade** is called as part of the initial Reader startup sequence, and it runs finitely. If it did not do so, it would cause the rest of the system to be delayed. If such a delays for more than several minutes, the Reader assumes a malfunction, and will proceed to reboot.

Custom Application upgrade program example

The following is an example of a Bash script **cust_app_upgrade**:

```
#!/bin/bash
event=$1

echo "Script cust upgrade starts"
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 Document 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 Octan
10/27/2010	4.6	Update revision for Octane 4.6
04/25/2011	4.8	Update revision for Octane 4.8, no chang
4/30/2012	4.10	Update revision for Octane 4.10, no chan
12/16/2014	5.2	Update revision for Octane 5.2 Updated to add xArray Gateway.
06/05/2015	5.4	Update revision for Octane 5.4 Updated to add SFTP Clarification and format updates
10/30/2015	5.6	Update revision for Octane 5.6

Date	**Revision**	**Comments**
12/21/2015	5.6.2	Update revision for Octane 5.6.2
10/10/2016	5.8.0	Update revision for Octane 5.8.0 - no changes
03/16/2017	5.12	Update revision for Octane 5.12 Minor change to the Dual Image Model table. Added Speedway R660 (xSpan) to supported list.
05/16/2017	5.12	Added Speedway R120 to supported list. Deprecated TFTP.

6 Notices

For more information, contact `support@impinj.com`