1

FWUpdate Module

Updating devices using GenApi

Version 1.0

Table of Content

Introduction

This document defines a generic way of carrying out firmware updates for GenICam-compliant devices.

Prior to this standard, firmware updates were carried out using proprietary update software provided by the device vendor. For transport layers like USB3 Vision such a vendor-specific update software relies on its own USB drivers being installed and bound to the device. This proved to be impractical when the device was used with third-party software and the associated third-party drivers.

This document standardizes the whole update process using the mechanisms provided by GenICam. It is therefore possible for any software vendor to update any device without installing the drivers of the device-vendor.

2 High Level Architecture

This standard specifies a format for a Generic Update File (GUF). Such an update file can contain multiple firmware updates for an arbitrary number of devices.

A single update contains a set of conditions to check if the update is suitable for the device. These conditions are regular expressions that are matched against features in the GenICam node tree of the device. If a regular expression matches, the corresponding condition is considered true.

The steps to physically execute the update are described in so-called procedures. Such a procedure consists of a sequence of steps that are executed unconditionally. There are five types of steps available providing a flexible way to interact with the GenICam features of the device.

3 Reference Documents

GenICam GenApi standard	Generic Interface for Cameras, version 2.1.1
GenICam SFNC	GenICam Standard Features Naming Convention, version 2.3
ETSI TS 102 918	Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC)
Camera Link	Camera Link® Standard
GigE Vision	GigE Vision® Standard
USB3 Vision	USB3 Vision® Standard
ISO 639-1/2	Standardized language codes.
PCRE	Perl Compatible Regular Expressions
POSIX "Fully portable filenames"	Portable Filename Character Set

4 GUF File Format

The GUF (Generic Update Format) file format is a container format for firmware updates. A single GUF file can contain any number of updates for any number of devices. The GUF file is a zip file with the following properties:

- It should have the file extension ".guf".
- The compression type must be "Store". This allows easy signature checks without temporarily unpacking the whole file.
- It must at least contain a zip file named "package.zip".
- If the GUF file is signed, it must comply with the ASiC-S Standard and use the CAdES Signature as described in ETSI TS 102 918. Accordingly, the GUF must contain a META-INF folder with exactly one file named "signature.p7s", containing the signature.
- If the GUF file is not signed, the META-INF folder must not exist.

4.1 package.zip

The package.zip must either use the "store" or "deflate" compression algorithm to maximize portability. It contains the actual firmware update files and exactly one file named "control.xml".

The package.zip file can contain any number of additional files or directories required for the update. To ensure maximum compatibility, filenames must comply with the POSIX "Fully portable filenames" standard, which means that only the following characters are allowed: A–Z a–z 0–9 . _ -

4.2 control.xml

The control.xml file describes all firmware updates contained in this GUF. The file must comply with the XML schema http://www.genicam.org/GenFwUpdate/GenFwUpdateSchema_Version_1_0.xsd.

The encoding of the file must be UTF-8.

The basic structure of the control.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<GufRuleSet
  xmlns=”http://www.genicam.org/GenFwUpdate/Version_1_0”
  xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
  xsi:schemaLocation="http://www.genicam.org/GenFwUpdate/Version_1_0
            http://www.genicam.org/GenFwUpdate/GenFwUpdateSchema_Version_1_0.xsd”
  SchemaMajorVersion="1"
  SchemaMinorVersion="0" >
     <Procedure Id="p1">…</Procedure>
     <Procedure Id="p2">…</Procedure>

     <Update ProcedureId="p1">
          <Info>…</Info>
          <Condition Feature="DeviceModelName">ModelName</Condition>
          <Condition Feature="Feature2">Value2</Condition>
     </Update>
     <Update>…</Update>
</GufRuleSet>
```

As shown in the example XML the root node is <GufRuleSet>. The root node contains at least one <Procedure> and at least one <Update> node.

Attributes of FWUpdateSet:

- SchemaMajorVersion: The major version of the schema file
- SchemaMinorVersion: The minor version of the schema file

4.2.1 Internationalization

<Entry> and <Description> elements can be localized using a "Lang" attribute. The value of this attribute must be an official ISO language code. The full list of codes is available at http://www.loc.gov/standards/iso639-2/php/code_list.php . If the requested language is available in ISO 639-1, the corresponding two character code must be used. Otherwise, the three-character codes from ISO 639-2 can be used. Country codes are not supported.

To ensure a well-defined behavior in all cases, even when the language code requested by the user is not available, the following precedence rule applies:

1. Lang="<language code requested by the user>"
2. No Lang attribute specified
3. Lang="en"

This means that either an element with no Lang attribute or one with Lang="en" must be present.

4.2.2 <Update>

An update node describes an update for one device. The update node contains one info node with information on the update and any number of condition nodes, to check if the update is applicable to the device.

Attributes:

- ProcedureId: The id of a procedure which must be executed to perform this update.

4.2.2.1 <Info>

The Info node contains a set of key-value pairs to describe the update.

Example:

```
<Info>
     <Entry Key="Description">This firmware implements the magic color feature</Entry>
     <Entry Key="Description" Lang="de">Firmware mit neuem Farbfeature</Entry>
     <Entry Key="Version">4.2.0</Entry>
     <Entry Key="VersionStyle">dotted</Entry>
     <Entry Key="VersionExtractionPattern ">v=([\.0-9]+)</Entry>
     <Entry Key="ReleaseNotes">Some notes</Entry>
     <Entry Key="ReleaseNotesURL">https://baslerweb.com/coolreleasenotes.html</Entry>
     <Entry Key="Xy.custom">Custom metadata from vendor Xy</Entry>
</Info>
```

4.2.2.1.1 Standardized Info Keys

The keys shown in the table below are standardized and therefore have a defined meaning. The ones where localization is allowed are marked with a dot in the "I18n" column. Mandatory entries are marked in the "Mandatory" Column.

Key	**Description**	**I18n**	**Mandatory**
Description	The description of the update. This should be a short text with no more than 250 characters. Newlines must be specified using \n.	●	●
Version	The version of the update. The format is not specified. It must be ensured that two binary different updates for one device always have different versions.	-	●
VersionStyle	The style of the version value. This is needed to actually interpret and sort the versions. Using this information an update software is able to inform the user if an update would actually be a downgrade or if it was already applied to the device. The following styles are officially specified: **dotted** The version consists of any number of parts separated by dots. If a part consists of decimal characters only, it is compared numerically, otherwise it is compared using strcmp(). This leads to the following ordering: 1.1.a < 1.10.a < 1.10.b <1.10.b.a **semver** The style as specified in "Semantic Versioning 2.0.0" (http://semver.org/).	-	-

VersionExtractionPattern	Regular expression to extract the device version from the DeviceFirmwareVersion node. The first matched group is used as result. This is needed for devices which encode more information than just the firmware version inside the DeviceFirmwareVersion node. The default value is: **^(.*)$**	-	-
ReleaseNotes	Release notes of the update. This is treated as plain text. Newlines must be specified using \n. Tabs should be replaced by spaces.	●	-
ReleaseNotesURL	A link to a webpage with more release notes. This webpage can contain addition details not contained in ReleaseNotes.	●	-
UserSetPersistence	Specifies, if the device persists user sets during the update. Possible values are: **“none”** All user sets are reset to default values during the update. **“full”** The device guarantees, that all user sets are persisted during the update.	-	-
SequencerSetPersistence	Specifies, if the device persists sequencer sets during the update. Possible values are: **“none”** All sequencer sets are reset to default values during the update. **“full”** The device guarantees, that all user sets are persisted during the update.	-	-
Vendor.Custom	Any custom entries can be added as metadata by prepending them with “Vendor.”. “Vendor” must be replaced with the name of the company creating the GUF file.	●	-

4.2.2.2 <Condition>

Condition elements can be used to limit updates to devices fulfilling the given conditions.

The content of the condition node must be a regular expression that is matched against the GenICam feature specified inside the Feature attribute. For all node types except enum nodes matching must be done against the string representation of the GenICam node.

In the case of an enum node, matching must be done against the string representations of all **implemented** enum entries. If matching succeeds on any entry, the condition is considered true. To check if a enum entry is implemented by the device `GenApi::IsImplemented(entry.GetAccessMode())` must be used.

If multiple conditions are contained in one update node, all conditions must be true for the update to be applicable to the device.

Example:

```
<Condition Feature="DeviceVendorName">Bas.*</Condition>
```

4.2.3 <Procedure>

A procedure node describes an update procedure for a device. Multiple update nodes can reference the same procedure node.

Attributes:

- Id: The id of the procedure. It must be unique within the XML file.

The procedure node must contain at least one step node. When the software executes the procedure, the steps are executed unconditionally in the order of appearance. A failure while executing a step results in a failed update. All step nodes can contain the following child nodes:

- <Description> The description of the step. It should be human readable and contain some meaningful data which can for example be put into a log file by the update software. This element is internationalizable and therefore may appear several times with unique lang attributes.
- <ExpectedDurationMs> The expected execution time of the step in milliseconds. This node is optional. The value has no influence on the actual execution of the update. It can be used inside the update tool to better estimate the overall progress of the update. If no duration is given it defaults to 1 ms.

4.2.3.1.1 <FeatureWrite>

Writes the value to a GenICam feature of the device.

Child nodes:

- <FeatureName> The name of the feature to write.
- <Value> The value to write to the feature specified by FeatureName. FromString() is used to write the value to the Node.
- <Verify> Either "true" or "false". This is passed to the call to FromString(). The default is true.

4.2.3.1.2 <FeatureExecute>

Executes the given GenICam feature. The implementation must repeatedly call IsDone() until either the timeout specified by MaximumExecutionTimeoutMs is reached or IsDone() returns true. If the GenICam node is not of type ICommand, the update fails.

Child nodes:

- <FeatureName> The name of the feature to execute.
- <MaximumExecutionTimeoutMs> After this timeout polling should stop and the update fails.

4.2.3.1.3 <FileUpload>

Uploads a file contained in package.zip to the device using the File Access Control features

as defined in the GenICam SFNC.

Child nodes:

- <FileName> The path of the file inside package.zip. The file is allowed to live in subdirectories. Subdirectories are separated by a forward slash (/). The filename must not start with a slash.
- <DeviceFileName> The name of the file in the device.

4.2.3.1.4 <FeatureAssert>

Asserts that the given feature matches a given regular expression. If the assertion fails, the whole procedure and therefore the update fails.

Child nodes:

- <FeatureName> The GenICam feature to assert.
- <AssertPattern> A regular expression that is matched against the value returned from the ToString() method of the feature. If the match is empty, the assertion fails. If the feature is an enum node, matching must be done against all enum entries for which `GenApi::IsImplemented(entry.GetAccessMode())` returns true.

4.2.3.1.5 <DeviceReset>

Resets the device. After the reset the device must automatically be rediscovered using the rules specified in chapter 6.

Child nodes:

- <DeviceDiscoveryDelayMs> The delay in milliseconds, the firmware update tool must wait before it starts to rediscover the device. This does however not guarantee that the device is definitely available after this period because that also depends on the environment (like DHCP servers etc.). Therefore polling should be used after the delay.

5 Regular Expression Dialect

The regular expression dialect allowed in this standard is the one implemented by PCRE2 (http://www.pcre.org/). This has the added benefit that public tools like https://regex101.com/ can be used to test the expressions. For maximum portability it is suggested to use only basic regular expression operators.

This standard contains no mechanism for stating matching options. Therefore, all regular expressions must be executed using the following matching options:

- Quantifiers are greedy
- No multiline matching
- Non-anchored matching
- Case sensitive matching
- Text is treated as UTF-8. This works also well for ASCII 7-bit and is the best bet for the future

6 Device Rediscovery

Within the update process it is necessary to reset and rediscover the updated device. To ensure a standardized update process that works across different implementations the rediscovery must happen according the rules given in this chapter.

Secure rediscovery of a device is very transport layer specific. In general it can be said that it's the responsibility of the transport layer to ensure that the same device is rediscovered. For point-to-point transport layers like Camera Link® no further standardization is done.

6.1 TL Specifics

6.1.1 GigE Vision:

The MAC address of the device must be used to rediscover the device. Therefore, a firmware update must **never** change the MAC address of the device.

6.1.2 USB3 Vision:

The USB3 Vision GUID as specified in the USB3 Vision Standard must be used to rediscover the device.

7 Document History

Version	Date	Editor	Description of Changes
Draft A	2016-09-23	Stefan Klug, Basler AG	- First draft
0.1	2016-10-27	Stefan Klug	First internal Version
0.2	2016-11-04	Stefan Klug	Removed Procedure → Description Removed Procedure → Steps Added I18n Added Verify to FeatureWrite Fixed some typos
0.2.1	2016-11-11	Stefan Klug	Fixed some typos Added non-anchored matching requirement Specified mandatory Info Entries
0.3	2016-11-16	Stefan Klug	Special handling of Enum nodes in Conditions and FeatureAssert Renamed ETA to Duration Renamed attribute lang to Lang
0.3.1	2016-12-09	Stefan Klug	Text corrections
0.4.0	2017-01-26	Stefan Klug	Removed UserSetBackup/UserSetRestore Added UserSetPersistence and SeqencerSetPersistence info keys Changed name to Firmware Update Module Removed Question regarding UserSetBackup
0.5.0	2017-02-20	Stefan Klug	Renamed Duration to ExpectedDuration to be more precise Added EnumerationDelay to the ResetDevice step
0.6.0	2017-05-07	Stefan Klug	Clarify what happens when feature assert fails. Add MaximumExecutionTimeoutMs to FeatureExecute Change EnumerationDelay to DeviceDiscoveryDelayMs Specified that regex matching should be case sensitive More specs on the FileUpload::FileName Node
1.0RC1	2017-08-16	Stefan Klug	New official name “FWUpdate Module” Changed xsd naming for version 1
1.0RC2	2017-10-13	Stefan Klug	Fixes in refernce xsd and sample xml
1.0RC3	2018-03-09	Hartmut Nebelung	Adopt naming scheme to GenApi module
1.0	2018-04-06	Stefan Klug	Text corrections Improved “Referenced Documents” section Fixed xsd url Changed ExpectedDuration to ExpectedDurationMs Removed empty Q&A section